




DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

# AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/88D-8



# A REAL-TIME SIMULATOR FOR MAN-IN-THE-LOOP TESTING OF AIRCRAFT CONTROL SYSTEMS (SIMTACS-RT) THESIS

Gary G. Dameron
Captain, USAF

AFIT/GE/ENG/88D-8



Approved for public release; distribution unlimited

AFIT/GE/ENG/88D-8

# A REAL-TIME SIMULATOR FOR MAN-IN-THE-LOOP TESTING OF AIRCRAFT CONTROL SYSTEMS (SIMTACS-RT)

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

Gary G. Dameron, B.S.E.E.
Captain, USAF

December, 1988



Approved for public release; distribution unlimited

## Acknowledgement

The successful completion of this thesis, indeed the successful completion of AFIT, represented not only hard work on my part, but generous assistance and support from so many.

I am most grateful to my wife, Carol, for shouldering a tremendous load. Her willingness to take on extra responsibilities while I spent hours studying and working, and her moral support were vital to my success. She helped me maintin faith in myself, and my faith in God, which enabled me to keep a proper perspective.

Special thanks to my thesis advisor, Lt Col Zdzislaw Lewantowicz, whose unwavering faith in my abilities made possible much more progress in this effort than I had ever thought possible. His enthusiasm for this project brought out my best.

The biggest contributor to my day-to-day efforts was Mr. Don Smith, the SIMSTAR lab technician. His tremendous efforts to keep SIMSTAR running, his impromptu tutoring in the fine points of hybrid computing, and his patient explanations and longsuffering re-explanations were invaluable. As I said so often, "Don, you're a genius."

Finally, I'd like to thank my lab partner, Daryl Hammond, a great friend and confidant throughout the entire AFIT "ordeal," who even managed to negotiate a truce between me and computers. Thanks, Daryl.

## Table of Contents

## List of Figures

## List of Tables

## NOTATION

| SYMBOL | DEFINITION | UNITS |
|---|---|---|
| a | sonic velocity | ft/sec |
| ADOT | Change in angle of attack | deg/sec |
| AFIT | Air Force Institute of Technology | |
| AOA | Angle of attack | deg |
| AW | Aircraft weight | lb |
| $A_y$ | Lateral acceleration at the accelerometers | g's |
| $A_{y_{cg}}$ | Lateral acceleration through the center of gravity | g's |
| b | Wing span (30.0 ft) | ft |
| $\bar{c}$ | Wing reference chord (11.32 ft) | ft |
| $C_D$ | Drag coefficient with control surfaces in nominal position (F-16) | |
| $C_{D_{TOTAL}}$ | Total Nondimensional Drag | |
| CG | Center of gravity given as ratio of the difference of the aerodynamic center and center of gravity to the wing reference chord (CG=.35) | |
| $C_L$ | Lift coefficient with control surfaces in nominal position (F-16) | |
| $C_{L_q}$ | Lift to pitch rate derivative | per rad |
| $C_{L_{\dot{\alpha}}}$ | Lift to time rate of change in angle of attack derivative | per rad |
| $C_{L_{\delta FA}}$ | Lift to flaperon deflection derivative | per deg |
| $CL_{TOTAL}$ | Total Nondimensional Lift | |
| $C_l$ | Rolling Moment coefficient with control surfaces in nominal position (F-16) | |

| | | |
|---|---|---|
| $C_{l_p}$ | Stability axis rolling moment to roll rate derivative | per rad |
| $C_{l_r}$ | Stability axis rolling moment to yaw rate derivative | per rad |
| $C_{l_{\delta_{FA}}}$ | Rolling moment to differential flaperon deflection derivative | per deg |
| $C_{l_{\delta_{HA}}}$ | Rolling moment to differential horizontal tail deflection derivative | per deg |
| $C_{l_{\delta_r}}$ | Rolling moment to rudder deflection | per deg |
| $C_{l_{TOTAL}}$ | Total Nondimensional Rolling Moment | |
| $C_M$ | Pitching Moment coefficient with control surfaces in nominal position (F-16) | |
| $C_{M_q}$ | Pitching moment to pitch rate derivative | per rad |
| $C_{M_{\dot{\alpha}}}$ | Pitching moment to time rate of change in angle of attack derivative | per rad |
| $C_{M_{TOTAL}}$ | Total Nondimensional Pitching Moment | |
| $C_n$ | Yawing Moment coefficient with control surfaces in nominal position (F-16) | |
| $C_{n_p}$ | Stability axis yawing moment to stability roll rate derivative | per rad |
| $C_{n_r}$ | Stability axis yawing moment to stability yaw rate derivative | per rad |
| $C_{n_{\delta_{FA}}}$ | Yawing moment to differential flaperon deflection derivative | per deg |
| $C_{n_{\delta_{HA}}}$ | Yawing moment to differential horizontal tail deflection derivative | per deg |
| $C_{n_{\delta_r}}$ | Yawing moment to rudder deflection | per deg |
| $C_{n_{TOTAL}}$ | Total Nondimensional Yawing Moment | |
| $C_Y$ | Side Force coefficient with control surfaces in nominal position (F-16) | |

| | | |
|---|---|---|
| $Cy_p$ | Side force to stability roll rate derivative | per rad |
| $Cy_r$ | Side force to stability yaw rate derivative | per rad |
| $Cy_{\delta FA}$ | Side force to differential flaperon deflection derivative | per deg |
| $Cy_{\delta HA}$ | Side force to differential horizontal tail deflection derivative | per deg |
| $Cy_{\delta r}$ | Side force to rudder deflection derivative | per deg |
| $Cy_{TOTAL}$ | Total Nondimensional Side Force | |
| DAP | Digital Arithetic Processor- The SIMSTAR has one for executing the digital region | |
| DFCS | Digital flight control system | |
| DH | Symmetrical horizontal tail deflection ( also $\delta_H$ ) | deg |
| DLEF | Leading-edge flap deflection ( also $\delta_{LEF}$ ) | deg |
| Fa | Aileron pilot force stick input | lb |
| Fe | Elevator pilot force stick input | lb |
| Fr | Rudder pedal pilot input | lb |
| *g* | Acceleration of gravity | ft/sec/sec |
| h | Altitude | ft |
| H1 | Altitude | ft |
| $H_2$ | Altitude | ft |
| $I_e$ | Aircraft engine moment of inertia | slug-ft$^2$ |
| $I_x$ | Aircraft mass moment of inertia about the x body axis | slug-ft$^2$ |
| $I_{xy}$ | Aircraft mass product of inertia about the xy body plane | slug-ft$^2$ |
| $I_{xz}$ | Aircraft mass product of inertia about the xz body plane | slug-ft$^2$ |

| | | |
|---|---|---|
| $I_y$ | Aircraft mass moment of inertia about the y body axis | slug-ft$^2$ |
| $I_{yz}$ | Aircraft mass product of inertia about the yz body plane | slug-ft$^2$ |
| $I_z$ | Aircraft mass moment of inertia about the z body axis | slug-ft$^2$ |
| MACH | Mach number | |
| m | Aircraft mass | |
| $N_z$ | Normal acceleration at the accelerometers | g's |
| $N_{z_{cg}}$ | Normal acceleration through the center of gravity | g's |
| OHT | Original horizontal tail | |
| $p_o$ | Sea level static pressure (2116.216) | lb/ft$^2$ |
| $p_s$ | Stability axis roll rate | rad/sec |
| $p_S$ | Static pressure | lb/ft$^2$ |
| $q_s$, q | Stability axis and body axis pitch rate | rad/sec |
| $q_w$ | Wind axis pitch rate | rad/sec |
| $\bar{q}$ | Dynamic pressure | lb/ft$^2$ |
| $q_c$ | Total pressure (impact) | lb/ft$^2$ |
| r | Body axis yaw rate | rad/sec |
| $r_s$ | Stability axis yaw rate | rad/sec |
| $r_w$ | Wind axis yaw rate | rad/sec |
| $R_{L_q}$ | Flexible to rigid lift ratio for pitch rate and time rate of change in angle of attack | |
| $R_{L_{\delta_H}}$ | Flexible to rigid lift ratio for horizontal tail deflection | |
| $R_{l_p}$ | Flexible to rigid ratio for the rolling moment to roll rate derivative | |

| | | |
|---|---|---|
| $R_{l_\beta}$ | Flexible to rigid rolling moment ratio for addition of the vertical tail | |
| $R_{l_{\delta a}}$ | Flexible to rigid rolling moment ratio for defelction of total roll control for a gearing of 0.25 degree δHA for each 1 degree of δFA | |
| $R_{l_{\delta FA}}$ | Flexible to rigid rolling moment ratio for deflection of differential flaperons | |
| $R_{l_{\delta r}}$ | Flexible to rigid rolling moment ratio for rudder deflection | |
| $R_{M_q}$ | Flexible to rigid pitching moment ratio for pitch rate and time rate of change in angle of attack | |
| $R_{M_{\delta H}}$ | Flexible to rigid pitching moment ratio for addition and deflection of the horizontal tail | |
| $R_{n_{rVT}}$ | Flexible to rigid yawing moment ratio of the vertical tail for the yaw rate derivative | |
| $R_{n_\beta}$ | Flexible to rigid yawing moment ratio for the addition of the vertical tail | |
| $R_{n_{\delta r}}$ | Flexible to rigid yawing moment ratio for rudder deflection | |
| $R_{Y_{rVT}}$ | Flexible to rigid side force ratio of the vertical tail for the yaw rate derivative | |
| $R_{Y_\beta}$ | Flexible to rigid side force ratio for the addition of the vertical tail | |
| $R_{Y_{\delta r}}$ | Flexible to rigid side force ratio for rudder deflection | |
| PSP | Parallel Simulation Processors - SIMSTARs can have up to two for running the analog portion of the machines. | |
| S | Wing area (300 sq ft) | $ft^2$ |
| TH | Thrust | lb force |
| T | Sampling period | sec |

| | | |
|---|---|---|
| $V_T$ | Wind X-axis velocity | ft/sec |
| $\alpha$ | Angle of attack | rad |
| $\dot{\alpha}$ | Time rate of change in angle of attack | rad/sec |
| $\beta$ | Sideslip angle | rad |
| $\delta_F$ | Symmetrical Flaperon deflection | deg |
| $\delta_{FA}$ | Differential Flaperon deflection | deg |
| $\delta_{FL}$ | Left Flaperon surface deflection | deg |
| $\delta_{FR}$ | Right Flaperon surface deflection | deg |
| $\delta_H$ | Symmetrical horizontal tail deflection (also DH) | deg |
| $\delta_{HA}$ | Differential horizontal tail deflection | deg |
| $\delta_{HL}$ | Left horizontal tail surface deflection | deg |
| $\delta_{HR}$ | Right horizontal tail surface deflection | deg |
| $\delta_{LEF}$ | Leading-edge flap deflection (also DLEF) | deg |
| $\delta_r$ | Rudder deflection | deg |
| $\Delta C_{L_{FLEX}}$ | Incremental lift coefficient due to flexibilizing the wing/body | |
| $\Delta C_{L_{FT}}$ | Incremental lift coefficient due to full scale effects | |
| $\Delta C_{M_{FLEX}}$ | Incremental pitching moment coefficient due to flexibilizing the wing/body | |
| $\Delta C_{M_{FT}}$ | Incremental pitching moment coefficient due to full scale effects | |
| $\theta$ | Body axis pitch Euler angle | rad |
| $\Phi$ | Body axis roll Euler angle | rad |
| $\Psi$ | Body axis yaw Euler angle | rad |

| | | |
|---|---|---|
| $\rho$ | Air density | $slug/ft^3$ |
| $\rho_o$ | Air density at sea level | $slug/ft^3$ |

## Abstract

A real-time, high-fidelity simulator is constructed to model F-16 dynamics and control laws. Built around an Electronics Associates Incorporated (EAI) SIMSTAR hybrid computer, the simulator (SIMTACS-RT) uses non-linear, coupled differential equations for its dynamic model. An EAI FGS 300 function generator is used to access an aerodynamic data base of 25000 values in real time. Man-in-the-loop simulation is supported with a force stick for pilot inputs and an oscilloscope display for pitch and roll information (the lateral program is still in development). Four hybrid computer programs are presented as user-ready simulation/analysis tools, supporting both multi-rate digital and analog control laws. Recommendations for further improvement in simulator realism are presented.

# A REAL-TIME SIMULATOR FOR MAN-IN-THE-LOOP TESTING OF AIRCRAFT CONTROL SYSTEMS (SIMTACS-RT)

## I. Introduction

### Overview

Control sytems for modern high-speed, high-maneuverability aircraft are complex. It is imperative that these control systems be thoroughly tested for pilot compatibility prior to their implementation on actual aircraft. The subject of this thesis is the design and implementation of a high fidelity, real-time simulator which is intended for use in the design of control systems. Using a hybrid computer as its processor, the system is designed to be easily reconfigurable to a variety of aircraft and controller designs.

### Background

As part of the guidance and control curriculum at the Air Force Institute of Technology (AFIT), a great deal of academic study and laboratory experimentation is devoted to the design of control systems. These control systems are used in many applications including aircraft control, robotics, tracking, navigation and fire control. Controllers are designed using various analytic techniques. Testing a design's interaction with a human

operator is the final milestone to be completed prior to building and testing a prototype. Ultimately, a control system's real purpose is to relay commands from the user to the machine, and to modify the machine's dynamic characteristics if necessary.

In general, simulation of a complex control system has been difficult, especially if plant dynamics are non-linear and time-varying. Methods have been developed to estimate performance based upon linear approximations over small ranges of operation (3,6,16). Computer-aided design (CAD) programs, such as $MATRIX_X$ (19) or TOTAL (13) have been developed to assist in the evaluation of controller designs based upon linearization theory.

However, this method of design evaluation has drawbacks. Engineers need to be able to test controllers over large ranges of operation (2). Maneuvers such as those in aerial combat call on controllers to operate in rapidly changing conditions (2). Most importantly, real operating environments are non-linear (3:13-14).

Linearized controller design procedures permit only approximate models of the human user (16:Ch 12). Most models consist of terms such as gain, pure time delay, and lead-lag transfer functions. This human model approximation approach is adequate for a first design. This approach needs to be supplemented with numerous levels of

simulations in an attempt to provide realistic system performance with a human operator at the controls.

The designer would find it useful to have a high-fidelity simulator which can accommodate the non-linearities of the environment as well as a large range of operation throughout the design process. Such a tool could reduce the design effort. In his master's thesis Mark Kassan demonstrated that such a simulator can be developed with Electronic Associates Incorporated (EAI) SIMSTAR hybrid computers (12). As a first step toward this goal of a near-real-time simulator which allows man-in-the-loop testing, Kassan designed and implemented simulations of the General Dynamics F-16 fighter aircraft. This simulator provided valid test data even over a large range of maneuvers including non-linearities.

Problem Statement

The objective of this thesis is to continue the development of a high-fidelity simulator, using the SIMSTAR hybrid computer, for the testing of control systems. The simulator must operate in real-time with the highest dynamic and visual display fidelity possible to the operator, while providing valuable information to the engineer evaluating his control system design.

Summary of Current Literature

The design of controllers for high performance aircraft is covered in a large body of literature.

Controller design techniques include conventional, modern, stochastic, and quantitative feedback theory (QFT).

The conventional design approach requires analysis of the basic plant dynamics. Plant output is compared to desired, or commanded, input. The difference between output and the commanded value is used as an actuating signal to drive the system toward the desired output. For multiple-input, multiple-output (MIMO) systems, several of the output variables may be compared to the input. Feedback loops are closed individually, optimizing the role each feedback parameter plays in the overall system. Blakelock gives an excellent overview of this procedure in his text on aircraft control design (3:306-332).

Modern control design consists of assembling plant characteristic equations into matrices. These matrices are manipulated to develop a controller design. Such techniques lend themselves very well to digital computer-aided design (CAD) tool manipulation. This characteristic makes modern control theory a powerful tool in rapid optimization of linear control problems (5:14).

Stochastic design techniques take into account dynamic disturbances from the environment, and imperfect measurements from sensors (14:2-9). All avaialable information provided to the controller is used, and optimum responses are calculated.

QFT is a procedure uses Nichols chart graphical techniques. It takes into account variations in the plant dynamic response, for instance battle damage to a fighter aircraft control system. The QFT procedure designs simultaneously for a large range of variation in plant characteristics (10).

Reid and Etkin describe the importance of considering the human pilot in the design of control systems for aircraft(6:490-1). They state:

> Although the analysis and understanding of the dynamics of the airplane as an isolated unit...is extremely important, one must be careful not to forget that for many flight situations it is the response of the total system, made up of the human pilot and the aircraft, that must be considered...The care exercised in considering the human element in the closed-loop system made up of pilot and aircraft can determine the success or failure of a given aircraft design to complete its mission in a safe and efficient manner.

Simulation of aircraft and their control systems is an important step in their implementation and production. According to Col David Milam, deputy director of the National Aerospace Plane (NASP) program and a test pilot with vast flying and flight testing experience, ground simulator work should be done as early as possible and integrated into the design process. Ground simulation has already served to highlight potential control problems in the NASP. It has been Milam's experience that once the design process has reached the flight testing stage, only

minor changes made for safety reasons can be implemented(15). Some of the most elaborate flight simulators are in the class of the Large Amplitude Multi-mode Aerospace Research Simulator (LAMARS) motion-based simulator at the Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio (17); and the Real-Time Multi-Processor Simulator (RTMPS) project at NASA's Lewis Research Center, Cleveland, Ohio (1).

The value of simulation to the design of control systems thus established, this thesis presents the development of a high fidelity real-time simulator for use at AFIT. Chapter 2 discusses desired specifications for the simulator, SIMTACS-RT. Chapter 3 describes the simulated aircraft and flight control system, while Chapter 4 provides background in the mathematical model of physical systems. Next, Chapter 5 describes the construction and use of SIMTACS-RT. Specific conclusions and recommendations are covered in Chapter 6.

# II. Simulator Specifications

## Overview

The goal of this thesis is to develop a real-time simulator which will be used to test controller designs with a man in the loop. Developmental work on SIMTACS-RT has been based on simulation of an F-16A with a digital flight control system. Because an aircraft controller must operate in real time over a large range of flight conditions, it is important to simulate its characteristics with the highest fidelity possible to meaningfully evaluate its design. Simulator design is covered in this chapter.

## Simulation Objectives

Because SIMTACS-RT is intended to be a tool for evaluation of nearly-completed designs, its capabilities vary markedly from the capabilities of computer-aided design (CAD) programs already used for controller design.

CAD programs' simulation routines have two limitations.

1. Only point design can be easily evaluated.
2. Only non-real-time simulation is possible.

Control system design work is based on a dynamic model of the aircraft system at a particular point in the flight envelope. The aircraft nonlinear differential equations of

motion are linearized about that point. Aircraft data from that point are then used in the design process. The controller is designed, and simulations are performed using idealized inputs. A large number of individual simulations are required to cover a representative set of points throughout the flight envelope. This laborious approach may be adequate during early design. During the testing phase, however, it is useful to simulate flight over a continuum of conditions, as representative of actual aircraft flight (3:5-55,107-136).

CAD program simulation inputs are idealized inputs such as step, pulse, or impulse functions (19:SB4-1,6). A flight control system, acting as an interface between pilot and aircraft, is seldom asked to handle an idealized input. Instead, the controller must respond to a wide range of human pilot inputs over a variety of conditions such as takeoff, landing, cruise, aerial refueling, and combat. Additionally, these control inputs are applied continuously over a long periods of time. A controller's response to an idealized input may provide good responses in design simulation, but prove unacceptable in the actual aircraft (20). Improvement in providing this continuum, as well as the availability of data to support realistic pilot inputs is also important from the man-in-the-loop perspective. For the simulation to be acceptable, the pilot must "feel" that he is flying an actual

aircraft. The author's flight simulator experience, as well as numerous interviews with colleagues, indicates that the ability to fly continuously through a wide range of changing conditions is essential in providing realism.

Similarly, the visual display used by the pilot must correlate closely to one to which the pilot is accustomed, whether it is an instrument display or an outside environment simulation. The presentation of that display must be at a rate sufficiet to prevent time lags from modifying the simulator's dynamic response. It must not detract from the pilot's feel and must be sufficiently rapid to avoid distraction caused by flicker or jerkiness (4:220).

Ease of programming is an important factor in any test simulator. The engineer must be able to code and implement his aircraft aerodynamic model and controller design without spending a great deal of time learning the intricacies of a new coding procedure. For this reason, the program code must be modular in nature allowing quick loading, changing, and debugging by the engineer.

The objectives outlined above are the basis for this simulation project. It is necessary at this point to describe the equipment necessary to implement these goals.

## Equipment and Materials

The heart of the simulator is the SIMSTAR hybrid computer manufactured by Electronic Associates Incor-

porated. The SIMSTAR computer has the ability to perform simulations in both digital and analog domains and is designed for real-time simulation. Unlike earlier analog computers, which used patch panel programming, the analog portion of the SIMSTAR is programmed using high-level language program code statements. Connections are made by the computer internally through an electronic switch matrix. Scaling ranges are provided by the user and SIMSTAR internally scales its components (18:3-7).

The SIMSTAR consists of a Digital Arithmetic Processor (DAP), a Parallel Simulation Processor (PSP), and a Function Generation Processor (FGP) as shown in Figure 2.1. The DAP provides overall control of the simulation setup and performs on-line serial calculations. The PSP performs parallel (analog) continuous calculations. The FGP (EAI Model FGS 300) selects values from tabulated function data, interpolating between data points when necessary (18:53-58).

A continuous system simulation language (CSSL) is used to program the SIMSTAR. The basis for programming is the Advanced Continuous Simulation Language (ACSL) (22), developed by Mitchell and Gauthier Associates which provides on-line simulation capability. Its structure separates the program into initial set-up, digital, and parallel regions. The digital region of the program is compiled by a digital translation routine, D-TRAN, into

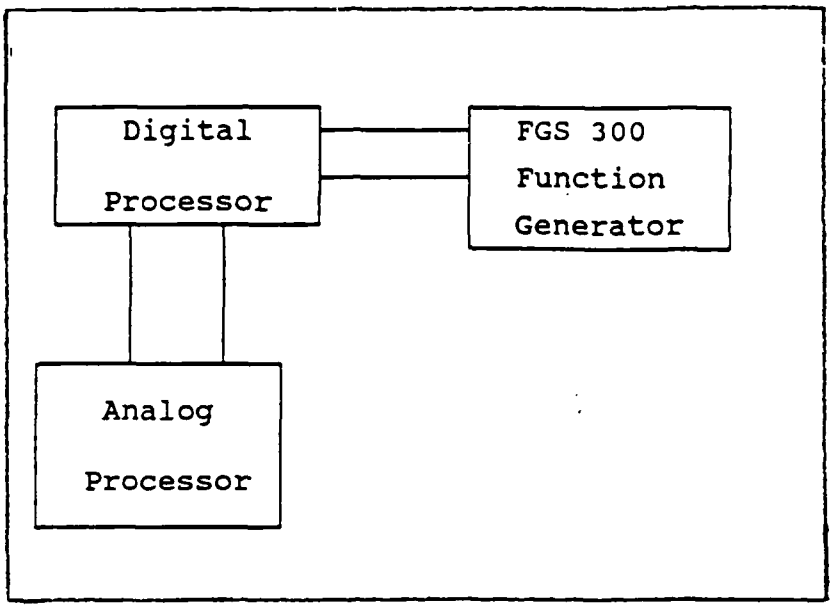



Figure 2.1 SIMSTAR Architecture

FORTRAN code for use in the DAP. The remainder of the program is compiled by the parallel translation routine, P-TRAN. Code compiled under P-TRAN provides instructions the DAP, enabling the DAP to internally connect mathematical computing blocks (MCB). The MCB's are the analog computation circuits. They are designed to perform the actual mathematical functions of integration, differentiation, summation, and multiplication (18:3-7).

Real-time aircraft simulator control inputs are provided by a force stick. The force stick is used to modulate a pure sinusoid source. The demodulated signals are the command control inputs to the aircraft controller model within the SIMSTAR. To provide similarity to control sticks used in fighter aircraft, SIMTACS-RT's

control stick is a force stick. The magnitude of the control signal it generates is proportional to the magnitude of the force applied to the stick. The control stick installed on SIMTACS-RT is very basic. To improve simulation realism, aircraft force control stick is needed

To provide visual cues to the pilot testing the simulated aircraft-controller combination, an oscilloscope display is presently used to provide a pitch bar and a roll bar display. This provides pitch and roll Euler angle information.

The graphical display must be improved if realistic pilot, controller interaction is to be simulated. The Air Force Flight Dynamics Laboratory is studying simulator visual displays and including heads-up instrumentation displays (HUD) using the Sun 3 workstation computerized graphics system. The graphics software developed by Flight Dynamics Laboratory can use simulator information to produce a realistic visual display (9). Follow-on work with SIMTACS-RT will interface with one of these graphics systems by feeding parameters indicating aircraft orientation to the graphics machine.

The set of F-16A flight data provided by the Air Force Flight Dynamics Laboratory, is explained in more detail in Chapter IV. It provides a basis of comparison of the simulator performance to the actual known aircraft

performance. The comparison demonstrates generic capabilities of SIMTACS-RT.(21)

Approach

SIMSTAR's analog processor solves the dynamic equations of motion and computes the aircraft orientation parameters. The analog processor can also be used to simulate an analog flight controller and aircraft engine dynamics. The function generation system contains aerodynamic data from F-16A flight and wind tunnel tests. The digital processor of the SIMSTAR can be used to simulate the digital flight controller used on the F-16A or other aircraft. In addition, SIMSTAR's digital processor performs calculations with aerodynamic coefficients provided by the FGS 300.

The SIMSTAR continuously calculates variables which describe the aircraft's position, velocity, orientation, and other states. The controller's inputs to the aircraft are calculated in the digital region and updated at a rate matching that of an actual digital controller. As the simulation progresses through the flight envelope and parameters change, the SIMSTAR uses the FGP to update the dynamic equation coefficients.

The simulator is validated by comparing the simulator's responses with data obtained from actual flight tests. After validating simulator dynamic response, pilot display upgrade can proceed.

## Scope and Limitations

The new function generator is a powerful addition to the SIMSTAR's simulation capabilities. It permits implementation of SIMTACS-RT on a single SIMSTAR computer making possible computational speeds very near real-time (18:53).

There are no limitations in the computation of aerodynamic data for the F-16A which would prevent high fidelity simulation. Customary simplifications for simulations are made(3:21). The aircraft model accounts for various aero-elastic effects due to angular velocities, control surface positions and sideslip angles. The current aircraft model does not account for ground effects, speedbrake and landing gear effects, engine dynamics and gyroscopic effects and variable mass for the aircraft. Although time did not permit implementation for this thesis, the model is easily expandable to include these additional flight conditions.

The model also incorporates assumptions of non-rotating Earth as an inertial reference frame and a homogenous atmosphere at rest with respect to Earth's surface. The aircraft is assumed to be symmetrical about a vertical plane defined by the longitudinal and vertical axes of the aircraft (12:16). The flight envelope modeled by the data is discussed in detail in Chapter IV.

Summary

The SIMSTAR computer system provides a powerful base for real-time high fidelity simulation of aircraft dynamics. Use of the F-16A data base available from Flight Dynamics Laboratory, along with integration of cockpit input and a visual display systems results in a simulator which is a valuable tool in the design of aircraft control systems.

# III. Aircraft and Flight Control System Description

## Overview

Although SIMTACS-RT has potential applications beyond flight control, the F-16A with a digital flight control system is used as a case study. The controller is a simplified version of a digital flight control system (DFCS) design obtained from the F-16 System Program Office (SPO) at Wright-Patterson AFB.

## Aircraft Description

The General Dynamics F-16A is a single-engine, single-seat, multi-role tactical fighter. The aircraft is powered by an F100-PW-100 turbofan engine in the 25,000 pound thrust class. The aircraft is modeled with the original horizontal tail (OHT), and carrying wingtip AIM-9 missiles.

The control surfaces consist of a pair of flaperons, a pair of horizontal stabilizers, a pair of leading-edge flaps, and a rudder. The flaperons are mounted on the trailing edge of each wing. Operated differentially they perform the function of ailerons. Moving in unison, they serve as flaps. The horizontal stabilizers mounted on the tail provide elevator and aileron functions through symmetrical and differential deflection. The leading-edge flaps are automatically controlled by the flight control

system to enhance performance over a wide speed range. The rudder, located on the vertical tail, provides directional stability and control.

The reasons for modeling the F-16 are two-fold. First, the aerodynamic data necessary for a high fidelity simulation was already available. Obtained from theF-16 SPO, the data is in a format suitable for use by the SIMSTAR/FGS 300 computer configuration. Lt Mark Kassan had transformed the data into a SIMSTAR digital file during his previous simulation work (12:19).

Second, the F-16 is a popular aircraft for study of flight control system design principles. The entire AFIT flight control curriculum laboratory design work is based on the F-16A. Additionally, several theses each year are based on the AFTI/F-16, a modified F-16A. The external modifications include the addition of twin vertical moveable canards on the forward underside of the engine inlet and a dorsal fairing on the upper fuselage center-line running from the back of the pilot's canopy to midway on the vertical fin root structure. All other external dimensions on the AFTI/F-16 are the same as the F-16A. Internally, the integrated servoactuators for the control surfaces are the same for both aircraft. With the exception of the canards, the simulator could be used to model the AFTI/F-16. Canards can be included in a future upgrade to the simulator (12:19).

Digital Flight Control System Description

The flight control system modeled is a simplified version of the Block 25 F-16 DFCS Functional Block Diagram (23). The full F-16 DFCS block diagram, divided into the longitudinal and lateral-directional modes of operation is shown in Figures 3.1 and 3.3, along with gain schedules in Figures 3.2 and 3.4. The control surface mixer is shown in Figure 3.5.

The flight control system, as currently modeled in the simulator, has several simplifications (12:20). The gain schedules as modeled, are based on flight at Mach 0.9 and an altitude of 20,000 feet in cruise configuration. This simplification was used for two reasons. First, the programming of the simulator is simplified during early validation stages. Programming an entire gain schedule for an aircraft uses multiple lines of source code and needlessly complicates modeling during the initial demonstration of the simulator. The program code is written in modular form to allow future detailing of the gain schedules to improve simulation fidelity.

Second, the classroom design projects undertaken at AFIT are based on a single design point, frequently the point Mach 0.9 and 20,000 feet. The single-point design, as discussed earlier, lends itself well to computer-aided design (CAD) methods. Students can complete their designs

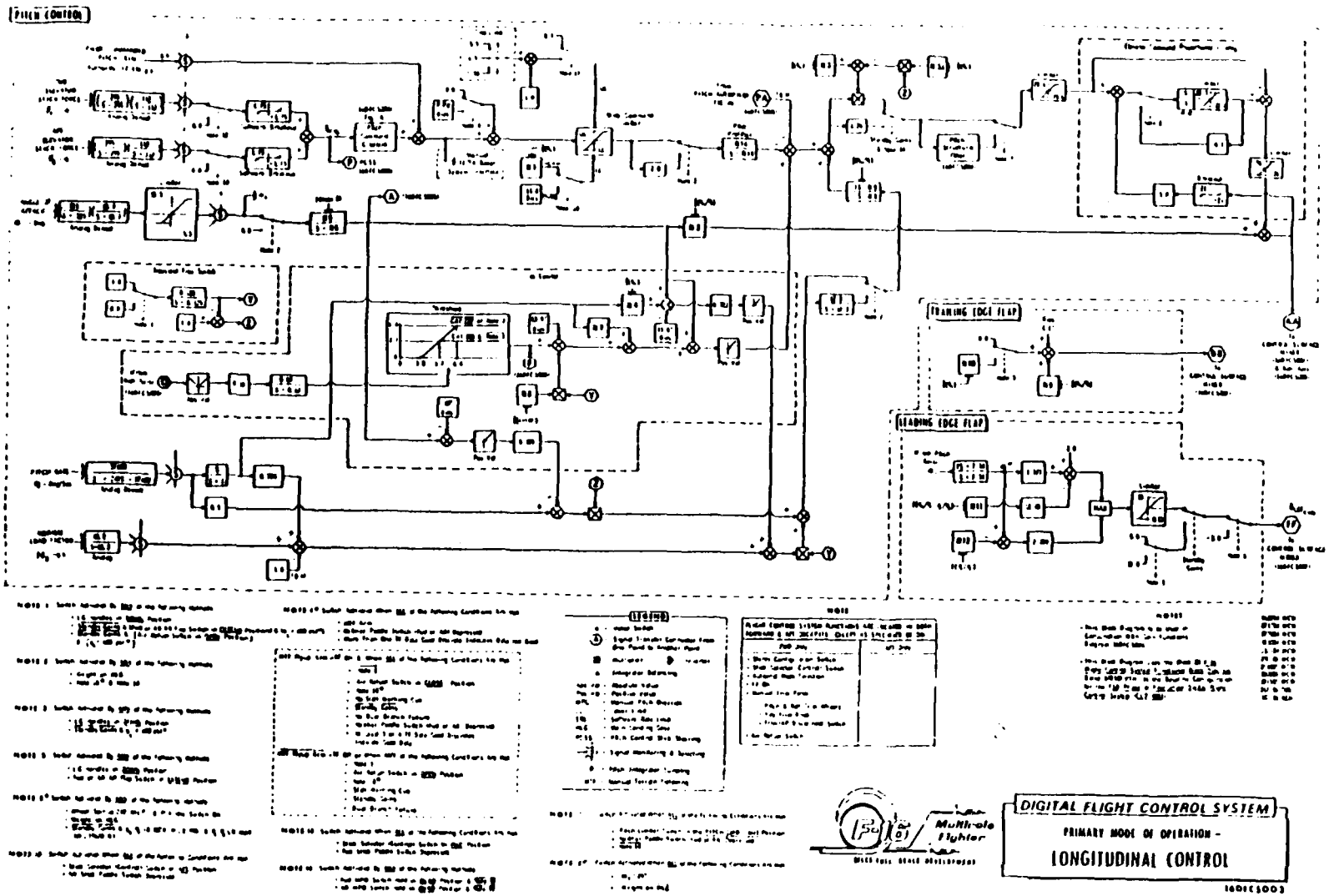



Figure 3.1 DFCS Longitudinal Control Block Diagram

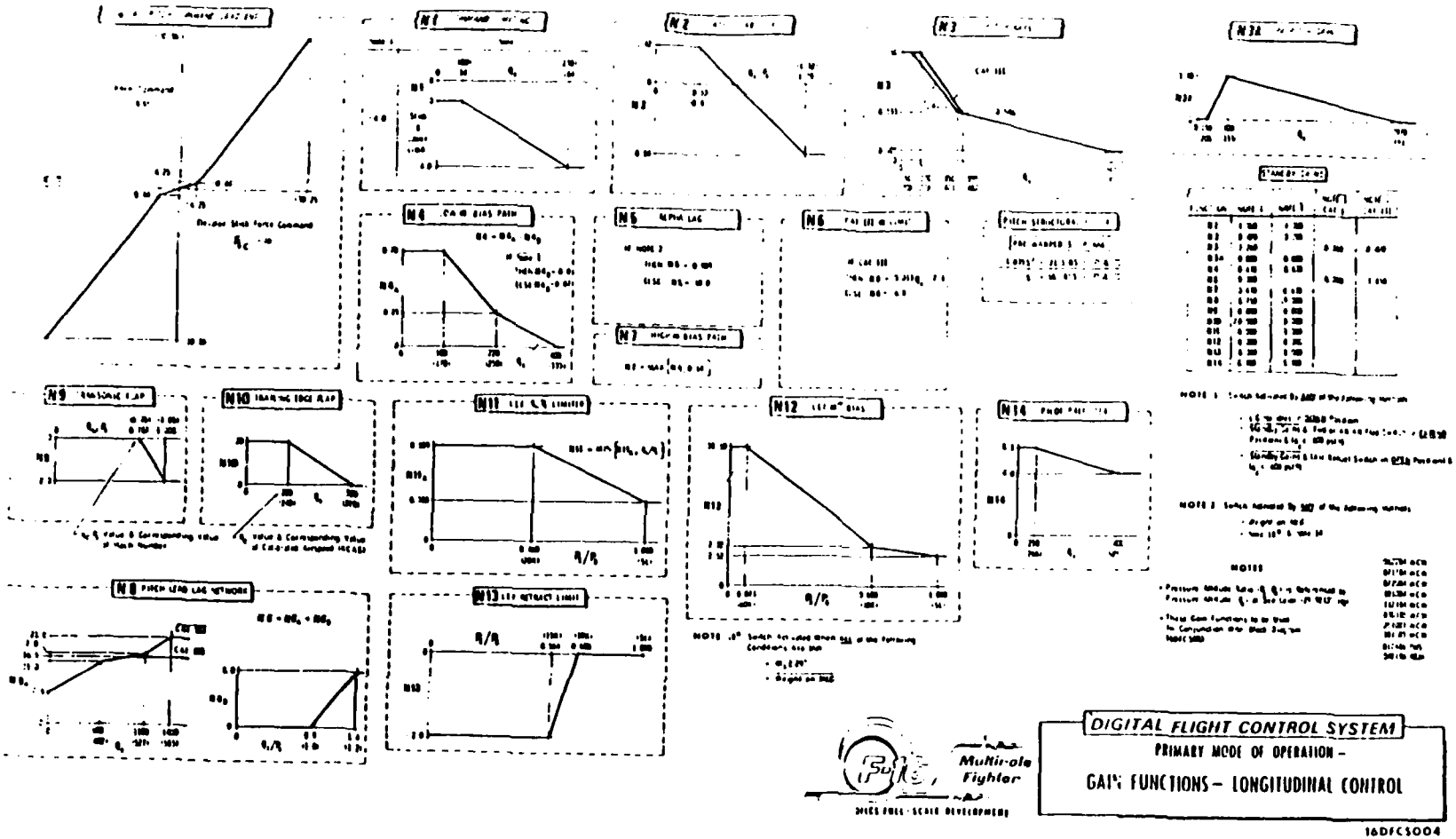



Figure 3.2 DFCS Gain Functions for Longitudinal Control

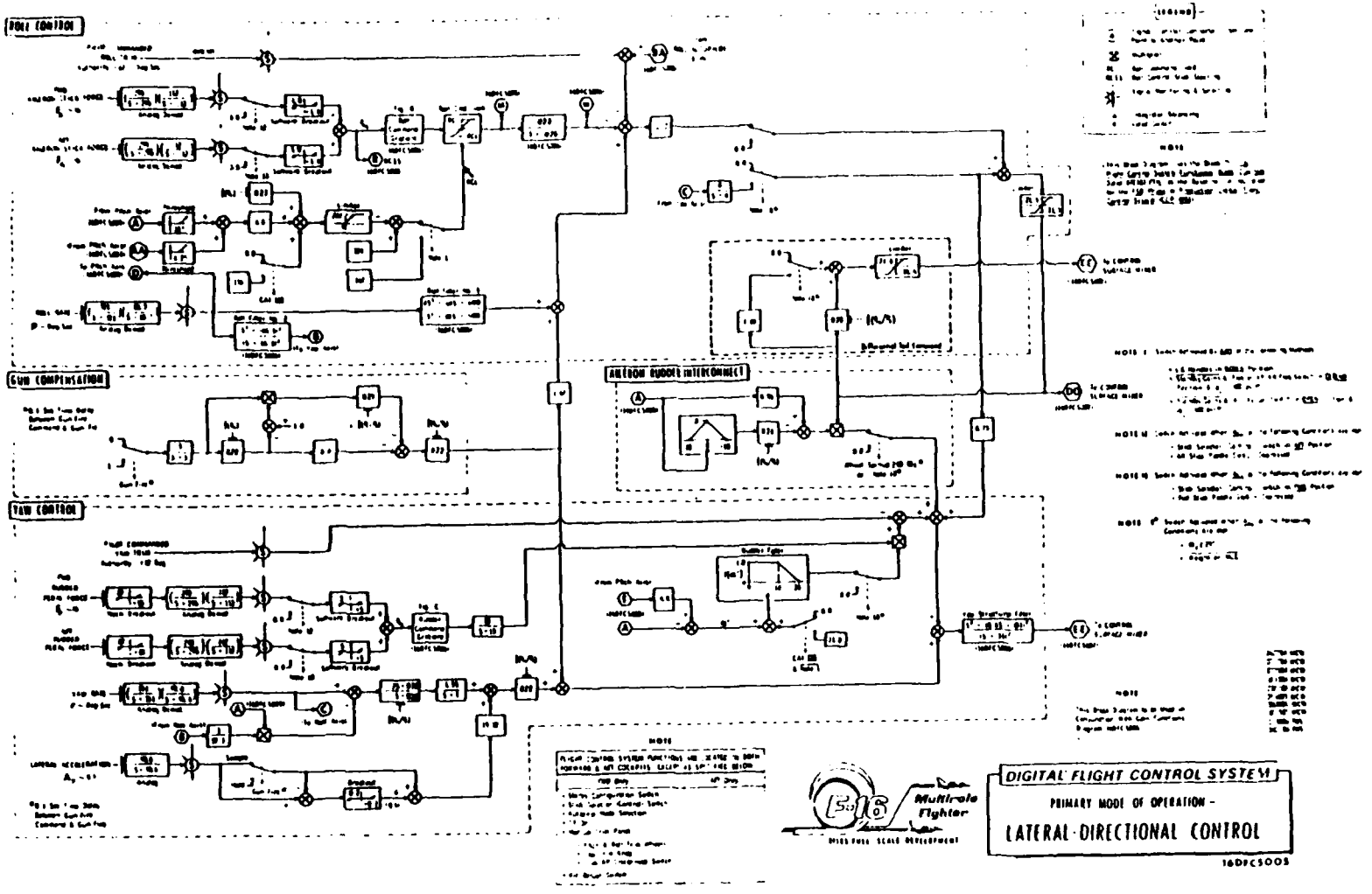



Figure 3.3 DFCS Lateral-Directional Control Block Diagram

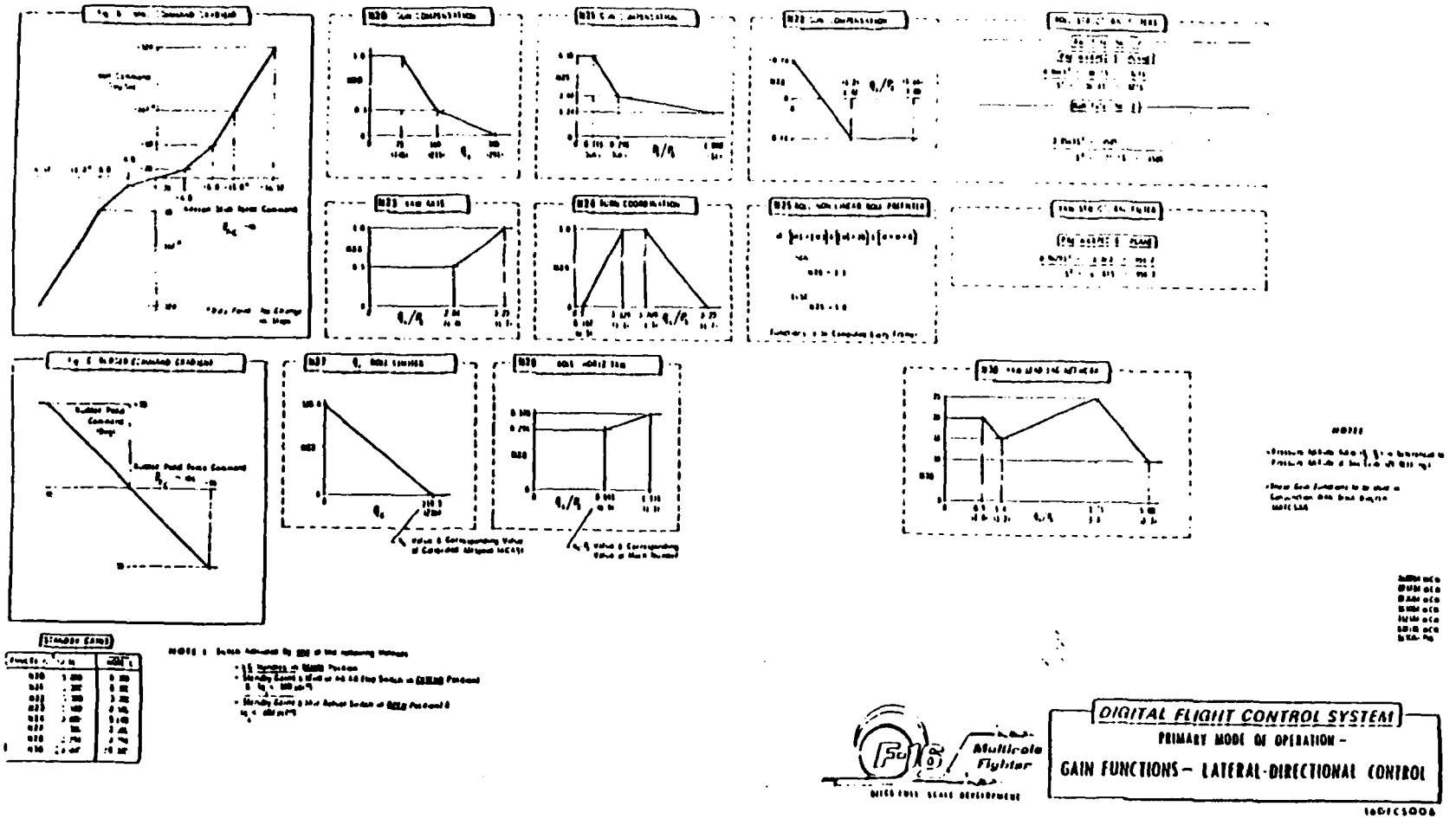



Figure 3.4 DFCS Gain Functions for Lateral-Direction Control

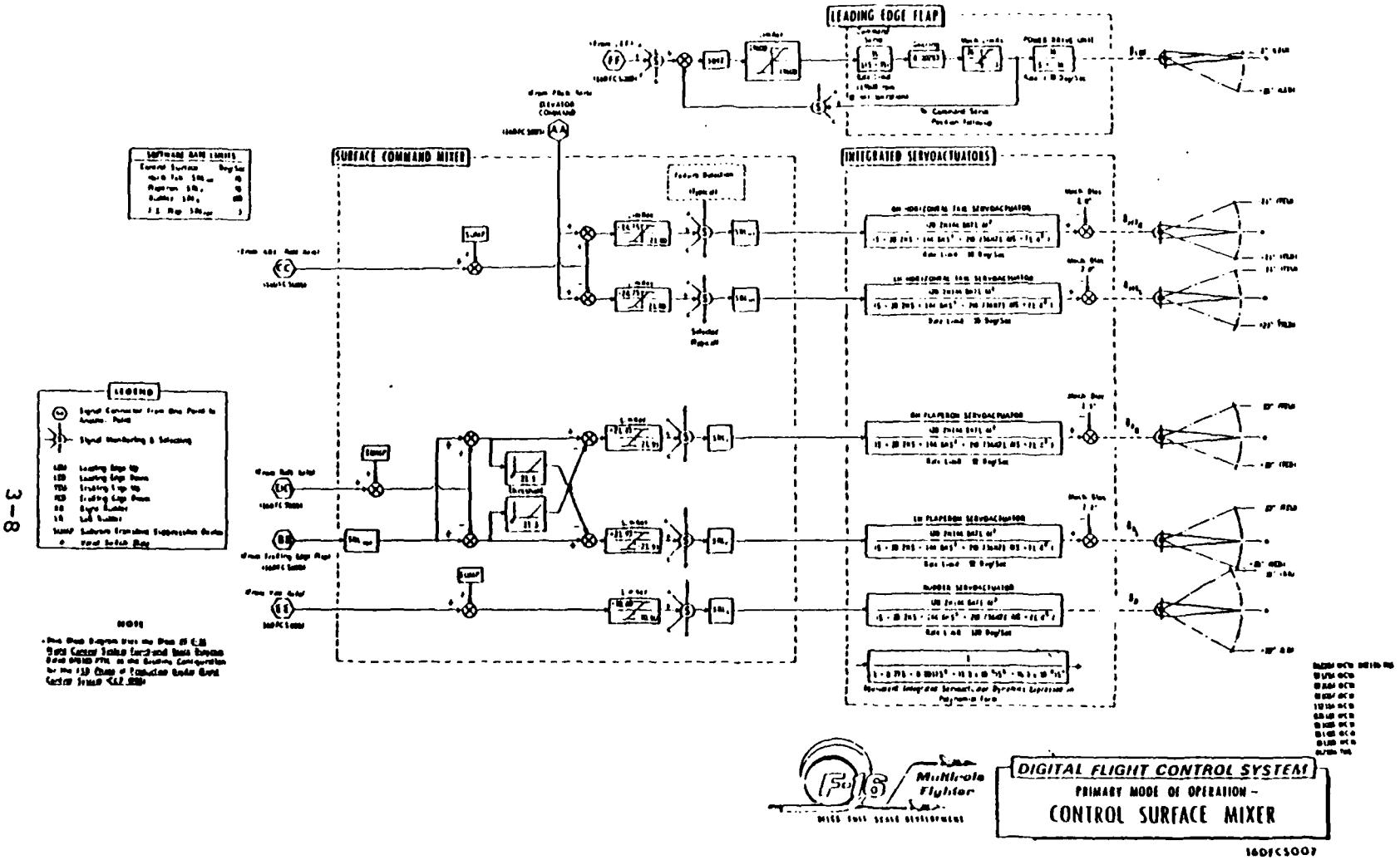



Figure 3.5 DFCS Control Surface Mixer

using the CAD programs, and transfer them directly to the simulator for validation.

The longitudinal DFCS uses pitch rate and normal acceleration feedback through a proportional-plus-integral controller. The controller operates through the forward loop providing elevator deflection. Angle of attack feedback provides static stability of the unstable F-16A aircraft (12:20).

The lateral-directional flight control system employs roll rate feedback to the aileron and differential elevator. A combination of lateral acceleration and yaw rate is fed back to the rudder and an aileron-rudder interconnect is used (12:20,26).

The simplified longitudinal and simplified lateral-directional flight control systems, as shown in Figures 3.6 and 3.7 respectively, are based upon the following simplifications and assumptions (12:26-31).

1. All gains are for Mach number of 0.9 and altitude of 20,000 feet. The gains are listed in Table I.

2. All mechanical and physical limiters are currently eliminated. If the aircraft is not overworked with high rates then this simplification is valid. Anti-wind-up limiters are also assumed unnecessary at this time.

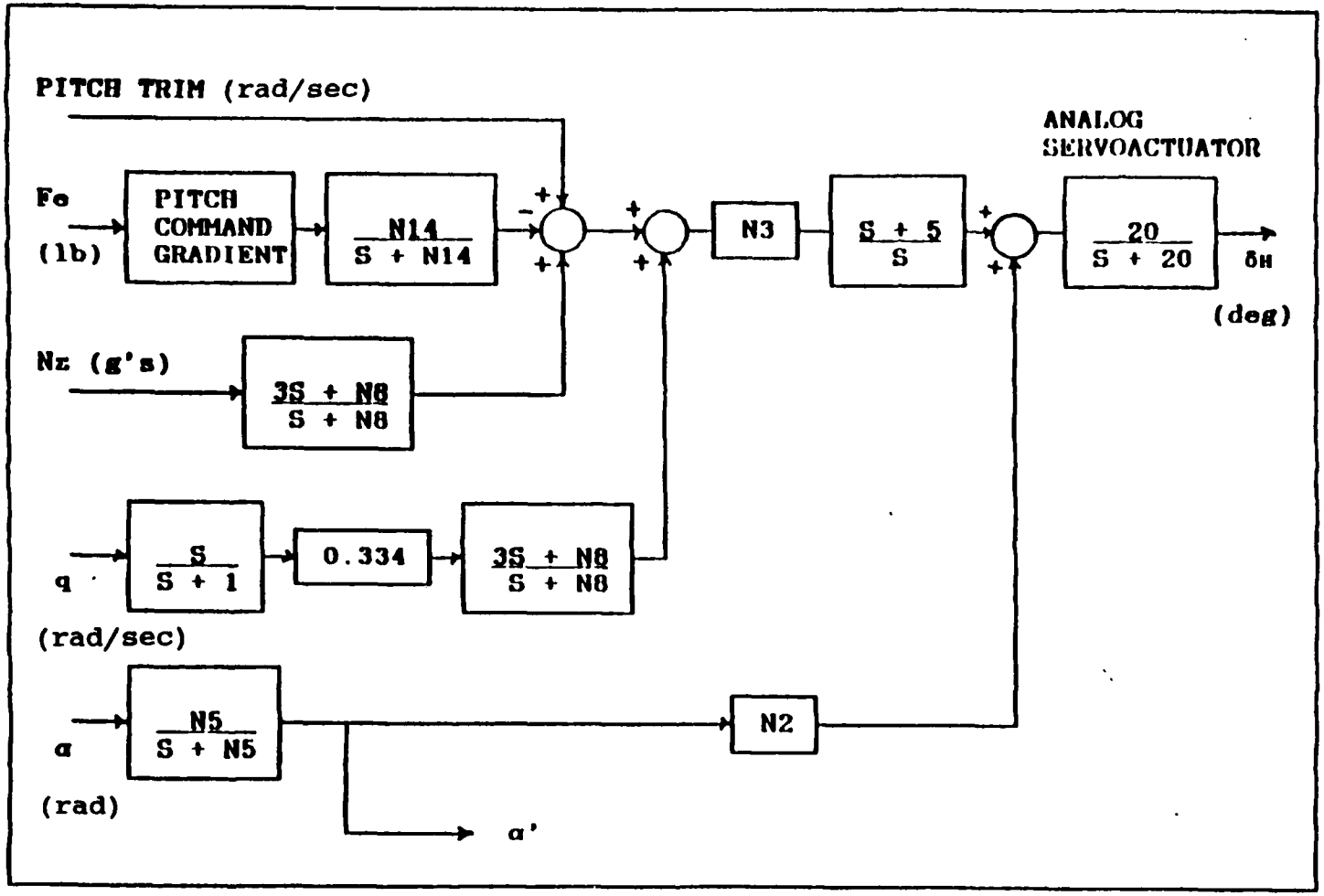



Figure 3.6. Simplified Longitudinal DFCS Block Diagram

Graphics adapted from (12).

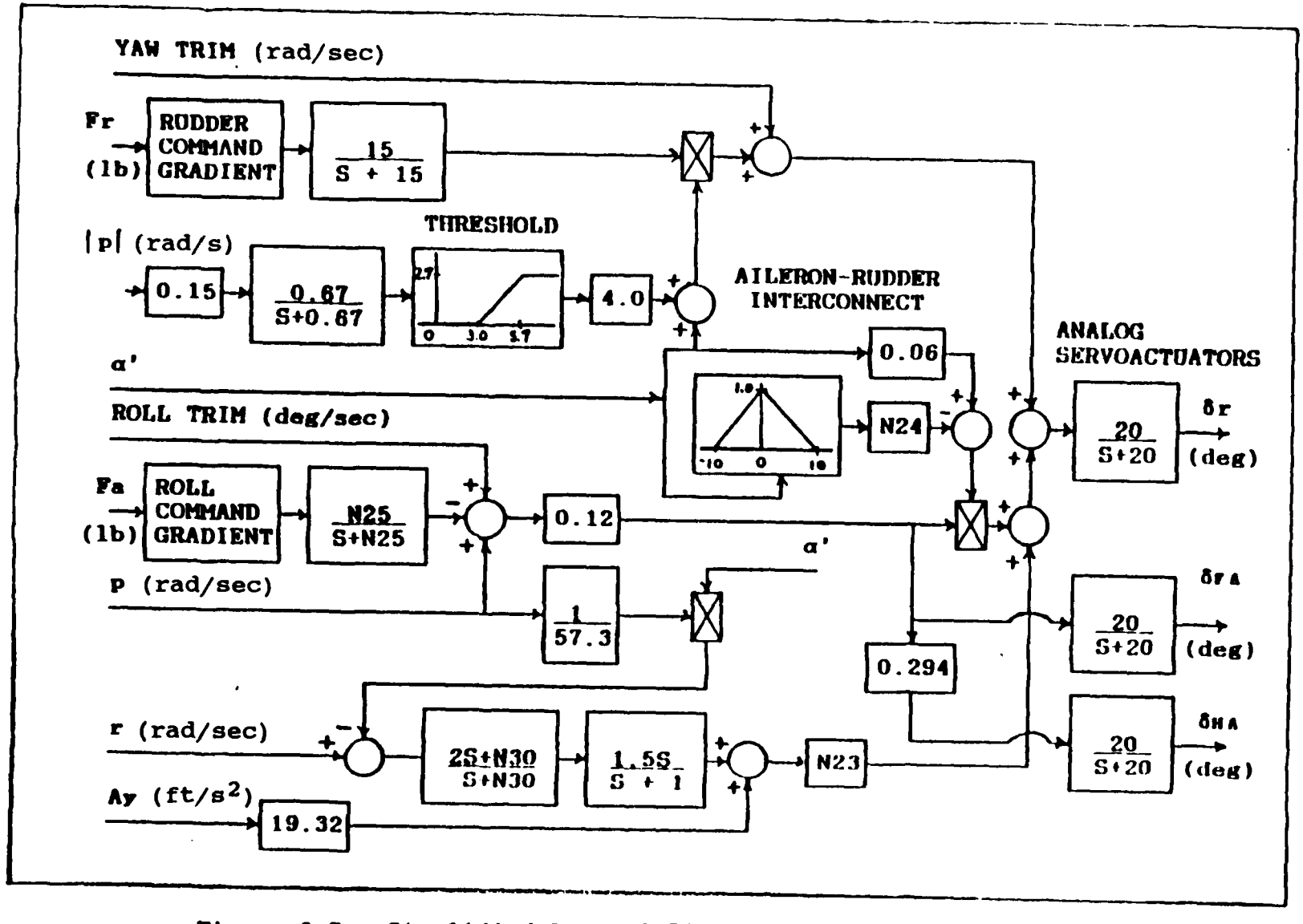



Figure 3.7. Simplified Lateral-Directional DFCS Block Diagram
Graphics adapted from (12).

Table I. Gains for Simplified DFCS

| Gain Constants | Gain Values |
|---|---|
| N2 | 0.38 |
| N3 | 0.70 |
| N5 | 10.00 |
| N8 | 14.40 |
| N14 | 7.20 |
| N23 | 0.50 |
| N24 | 0.67 |
| N25 | 2.50 |
| N30 | 20.00 |

3. The analog demodulators are removed. These are removed based on the assumption that the frequency content of the various signals will be relatively low.

4. The aircraft is considered to be flying with no extended landing gear or extended flaps and is not in an air refueling mode.

5. The servoactuators have been simplified to simple 20 rad/sec first-order lag filters which correspond to the lowest frequency filter of the actual servoactuators.

6. For lateral-directional mode, the gun compensation is ignored.

7. The yaw structural filter and the roll filters are eliminated. High frequency vibrations that these filters attenuate are considered insignificant for the purposes of this simulation.

8. The leading-edge flap deflection is considered zero for all time. This has little effect on the performance of the overall control system.

9. Any automatic angle of attack limiter loops are currently eliminated, simplifying the loop nesting of the block diagram.

10. The pitch command gradient for the simplified longitudinal controller is shown in Figure 3.8. The roll command gradient and the rudder command gradient for the simplified lateral-directional controller are shown in Figures 3.9 and 3.10.

The simplifications are performed without significant model degradation and provide a realistic controller for the specific flight condition and the surrounding flight envelope.

The next chapter discusses the modeling of the aircraft dynamics and flight control system.

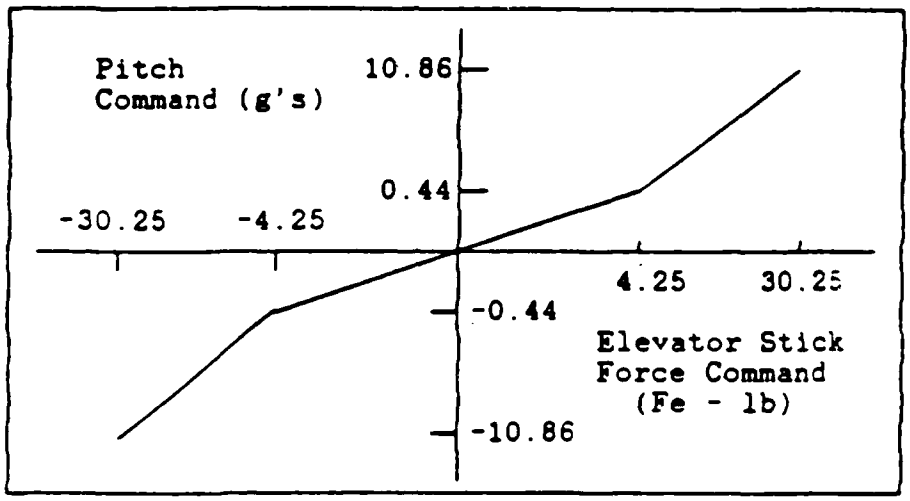



Figure 3.8. Pitch Command Gradient

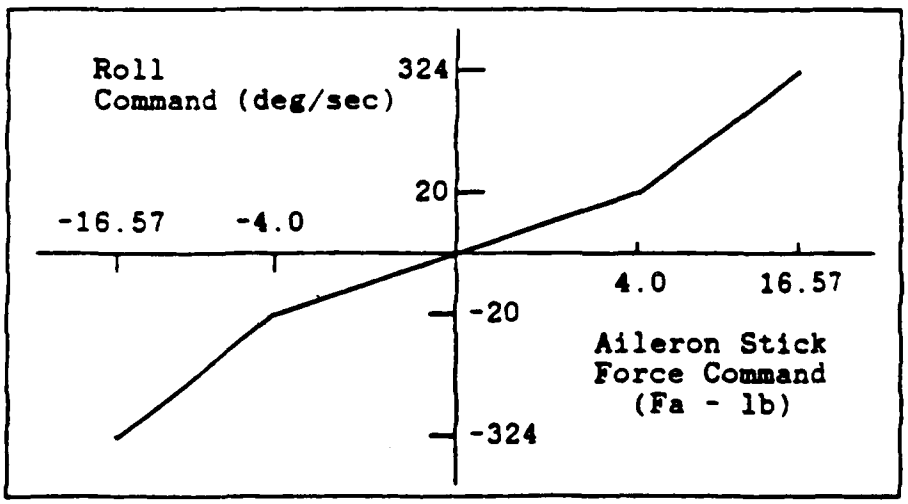



Figure 3.9. Roll Command Gradient
Graphics adapted from (12).

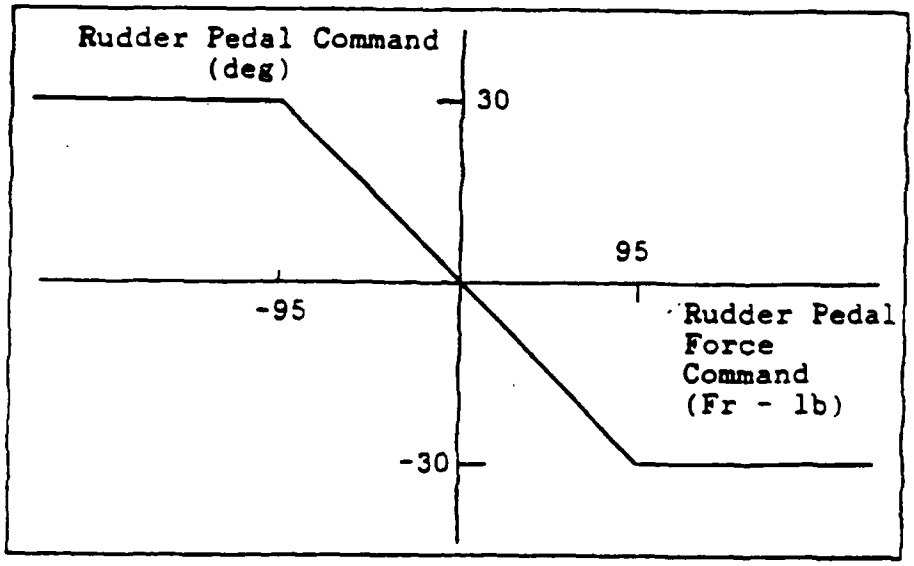



Figure 3.10. Rudder Command Gradient
Graphics adapted from (12).

# IV. Simulation Modeling

## Overview

The aircraft mathematical dynamic model is based upon the F-16A "clean" configuration aerodynamic data base obtained from ASD/ENFTC, Wright-Patterson AFB. It is identical to that used by Kassan (12).

The complete simulation model consists of equations of motion, atmospheric model, aerodynamic data format, aerodynamic data function generation, aerodynamic coefficient equations and backward difference equations for controller implementation. All equations are defined with respect to a set of sign and axis conventions.

## Axes and Sign Convention

The standard sign conventions and coordinate frames for the F-16 are as shown in Figure 4.1 and are summarized as follows:

$$\delta_F = \tfrac{1}{2} * (\delta_{FR} + \delta_{FL})$$

$$\delta_{FA} = \tfrac{1}{2} * (\delta_{FR} - \delta_{FL})$$

$$\delta_H = \tfrac{1}{2} * (\delta_{HR} + \delta_{HL})$$

$$\delta_{HA} = \tfrac{1}{2} * (\delta_{HR} - \delta_{HL})$$

The deflections $\delta_F$, $\delta_H$, $\delta_{HR}$, and $\delta_{FR}$ are defined positive with the trailing edge down and $\delta_{HL}$ and $\delta_{FL}$ are positive with the trailing edge up. The deflection $\delta_r$ is

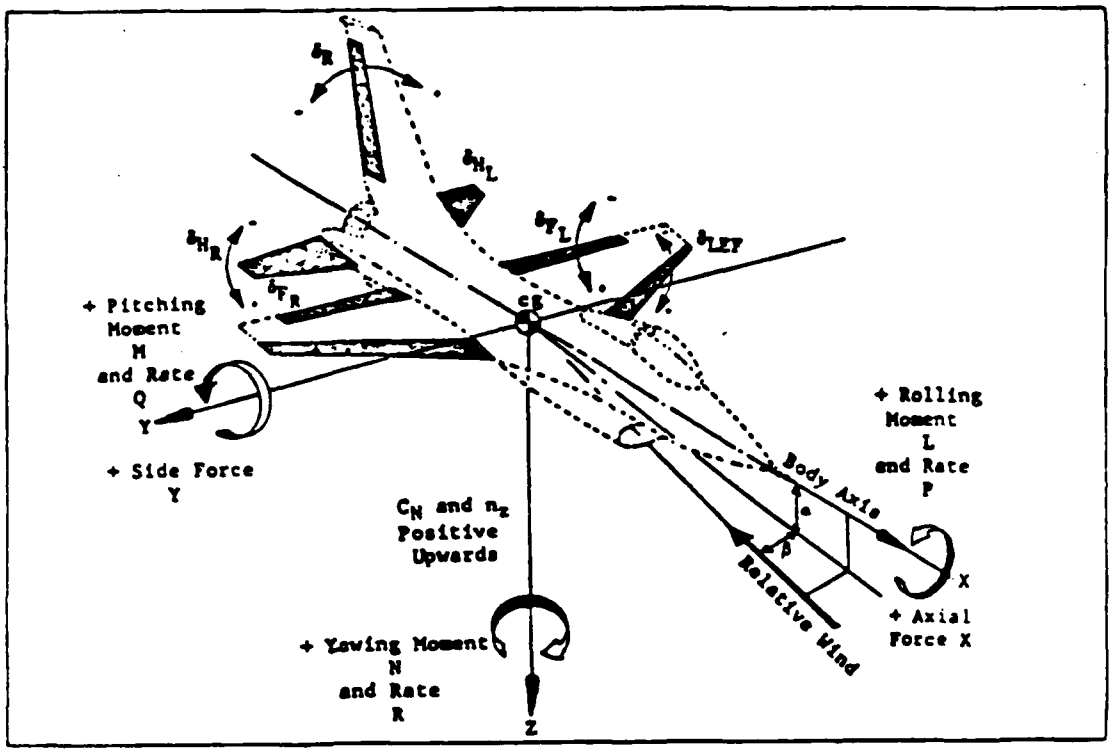



Figure 4.1. F-16 Sign Convention

positive with the trailing edge to the left of the aircraft.

Positive flaperon movement ($\delta_F$,$\delta_{FL}$, and $\delta_{FR}$) corresponds to additional lift on the corresponding wing. If the pilot moves the stick to the right, the left flaperon increases the lift on the left wing (positive $\delta_{FL}$), decreases lift on the right wing (negative $\delta_{FR}$) and the aircraft rolls to the right.

Positive rudder movement (positive $\delta_r$) refers to movement of the rudder trailing edge to the left, cor-

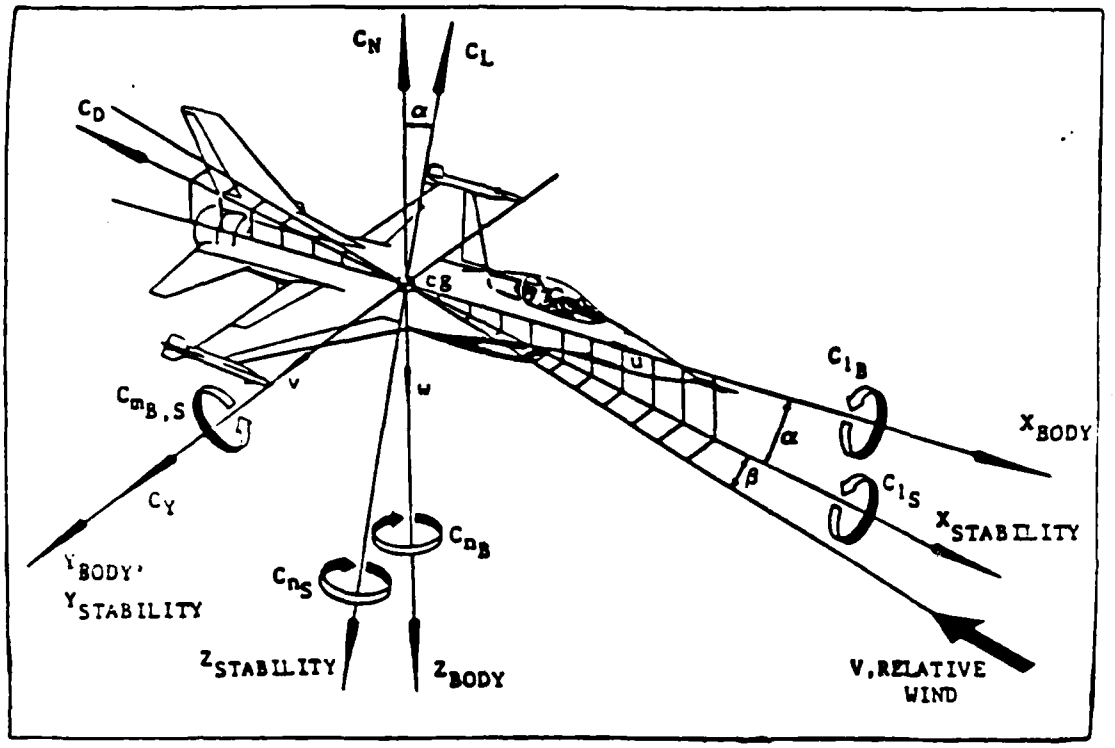



Figure 4.2. F-16 Axes System

responding to the pilot's pushing on the left rudder pedal and the aircraft nose points to the left.

The three axes used throughout this thesis are the stability, body, and wind axes. They are depicted in Figure 4.2. The aircraft center of gravity is the origin for all axes. Aerodynamic forces and moments are given in the stability axes. True forward velocity, $V_T$, is given in the wind axes.

Equations of Motion

The equations of motion for the aircraft are derived using basic kinematic and dynamic principles. The

equations provide a mathematical description of the forces, moments, velocities, and position of the aircraft. The equations provide this information in a variety of reference frames; body, wind, and Earth. They describe all six degrees of freedom.

Assumptions made for the purpose of this simulation are that the aircraft mass is constant, the nonrotating Earth is an inertial reference frame, and the atmosphere is at rest with respect to the Earth. The equations are taken from (21) unless otherwise noted.

Stability Axis Translational Forces. Inputs to stability axis translational force equations are lift, drag, and side force coefficients along with thrust (TH). The lift, drag, and side force coefficients are dimensionalized when multiplied by the dynamic pressure ($\bar{q}$) and the wing area (S). This dimensionalizing is done for all aerodynamic coefficients used in the equations of motion. The force due to gravity, as well as thrust, are generally given in the body axis. These are converted kinematically to the stability axis, making them usable with the aerodynamic coefficients.

$$X_S = (TH - AW\sin\theta)\cos\alpha + (AW\cos\theta\cos\phi)\sin\alpha - \bar{q}S\ C_{D_{TOTAL}} \qquad \text{(IV-1)}$$

$$Y_S = AW\cos\theta\sin\phi + \bar{q}S\ C_{Y_{TOTAL}} \qquad \text{(IV-2)}$$

$$Z_S = (AW\cos\theta\cos\phi)\cos\alpha - (TH - AW\sin\theta)\sin\alpha - \bar{q}S\ C_{L_{TOTAL}} \quad \text{(IV-3)}$$

where AW is aircraft weight.

Wind Axis Translational Forces. The previous three equations, converted to wind axes, are as follows:

$$X_W = X_S\cos\beta + Y_S\sin\beta \quad \text{(IV-4)}$$

$$Y_W = Y_S\cos\beta - X_S\sin\beta \quad \text{(IV-5)}$$

$$Z_W = Z_S \quad \text{(IV-6)}$$

Similarly, translational forces in the wind axis (Etkin 6:141) are also known to be:

$$X_W = \dot{m}\ V_T \quad \text{(IV-7)}$$

$$Y_W = m\ V_T\ r_W \quad \text{(IV-8)}$$

$$Z_W = -m\ V_T\ q_W \quad \text{(IV-9)}$$

Body Axis Moments. The moments are due only to the aerodynamic coefficients because all external forces on the aircraft act through the center of gravity. The moment equations are:

$$L = \bar{q}\ S\ b(C_{l_{TOTAL}}\cos\alpha - C_{n_{TOTAL}}\sin\alpha) \quad \text{(IV-10)}$$

$$M = \bar{q}\ S\ \bar{c}\ C_{M_{TOTAL}} \quad \text{(IV-11)}$$

$$N = \bar{q}\ S\ b(C_{l_{TOTAL}}\sin\alpha + C_{n_{TOTAL}}\cos\alpha) \quad \text{(IV-12)}$$

Body Axis Angular Accelerations. The full equations are shown here, but for the purposes of this thesis, the aircraft is assumed symmetrical and the longitudinal axis is in a plane of symmetry, thus the products of inertia

$I_{yz}$ and $I_{xy}$ are set to zero. Engine gyroscopic effects are ignored, so the engine moment of inertia, $I_e$, is set to zero.

$$\dot{p} = [L + (I_y - I_z)q\ r + I_{xz}(\dot{r} + p\ q) + I_{yz}(q^2 - r^2) + I_{xy}(\dot{q} - r\ p)]/I_x \quad \text{(IV-13)}$$

$$\dot{q} = [M + (I_z - I_x)r\ p + I_{xz}(r^2 - p^2) + I_{xy}(\dot{p} + q\ r) + I_{yz}(\dot{r} - p\ q) - I_e w_e r]/I_y \quad \text{(IV-14)}$$

$$\dot{r} = [N + (I_x - I_y)p\ q + I_{xz}(\dot{p} - q\ r) + I_{xy}(p^2 - p^2) + I_{yz}(\dot{q} + r\ p) + I_e w_e q]/I_z \quad \text{(IV-15)}$$

Integrating (IV-13), (IV-14) and (IV-15) yields the body axis angular velocities.

Stability Axis Angular Accelerations. The angular accelerations in the stability axes are used to find the aerodynamic angles. They are as follows:

$$p_s = p\cos\alpha + r\sin\alpha \quad \text{(IV-16)}$$

$$q_s = q \quad \text{(IV-17)}$$

$$r_s = r\cos\alpha - p\sin\alpha \quad \text{(IV-18)}$$

True Velocity and Aerodynamic Angles. From Equations (IV-7), (IV-8), (IV-9), derive the following relationships:

$$\dot{V}_T = X_w/m \quad \text{(IV-19)}$$

$$\dot{\alpha} = q_s - p_s\tan\beta + Z_w/(mV_T\cos\beta) \quad \text{(IV-20)}$$

$$\dot{\beta} = Y_w/(m\ V_T) - r_s \quad \text{(IV-21)}$$

Integrating Equations (IV-19), (IV -20) and (IV-21) yields the true velocity and aerodynamic angles.

Euler Angle Rates. The Euler angle rates are integrated to obtain the Euler angles. The Euler angles describe the rotation of the aircraft in the body axis with respect to the vehicle-carried axis. The vehicle-carried axis, or navigation frame, has its origin at the aircraft center of gravity, with X-axis pointing north, Y-axis pointing east, and z-axis pointing downward (NED) toward the origin of the Earth axes.

$$\dot{\psi} = (r \cos\phi + q \sin\phi)/\cos\theta \qquad \text{(IV-22)}$$

$$\dot{\theta} = q \cos\phi - r \sin\phi \qquad \text{(IV-23)}$$

$$\dot{\phi} = p + \dot{\psi} \sin\theta \qquad \text{(IV-24)}$$

Integrating Eqs (IV-22), (IV-23) and (IV-24) yield the Euler angles.

Body Axis Translational Velocities. The translational velocities in the body axes are needed for the calculation of the position and velocity of the aircraft with respect to the Earth.

$$u = V_T \cos\alpha \cos\beta \qquad \text{(IV-25)}$$

$$v = V_T \sin\beta \qquad \text{(IV-26)}$$

$$w = V_T \sin\alpha \cos\beta \qquad \text{(IV-27)}$$

Direction Cosines. The direction cosines are computed using the Euler angles of Equations (IV-22), (IV-23) and (IV-24). They are used to define the orientation of the body with respect to the Earth frames.

$$a_{11} = \cos\theta\cos\psi \quad \text{(IV-28a)}$$

$$a_{21} = \sin\theta\cos\psi\sin\phi - \sin\psi\cos\phi \quad \text{(IV-28b)}$$

$$a_{31} = \sin\theta\cos\psi\cos\phi + \sin\psi\sin\phi \quad \text{(IV-28c)}$$

$$a_{12} = \cos\theta\sin\psi \quad \text{(IV-28d)}$$

$$a_{22} = \sin\theta\sin\psi\sin\phi + \cos\psi\cos\phi \quad \text{(IV-28e)}$$

$$a32 = \sin\theta\sin\psi\cos\phi - \cos\psi\sin\phi \quad \text{(IV-28f)}$$

$$a_{13} = -\sin\theta \quad \text{(IV-28g)}$$

$$a_{23} = \cos\theta\sin\phi \quad \text{(IV-28h)}$$

$$a_{33} = \cos\theta\cos\phi \quad \text{(IV-28i)}$$

Earth Axis Velocities. The direction cosines from Equations (IV-28a) through (IV-28i) are used to convert the body axis translational velocities (u,v,and w) from Equations (IV-25), (IV-26) and (IV-27) into velocities in the Earth axes.

$$\begin{bmatrix} \dot{X}_e \\ \dot{Y}_e \\ \dot{Z}_e \end{bmatrix} = \begin{bmatrix} a_{11}u + a_{21}v + a_{31}w \\ a_{12}u + a_{22}v + a_{32}w \\ a_{13}u + a_{23}v + a_{33}w \end{bmatrix} = A \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad \text{(IV-29), (IV-30), (IV-31)}$$

Integrating Equations (IV-29), (IV-30) and (IV-31) yields the position of the aircraft in the Earth axes.

Miscellaneous Equations. The following equations are necessary to determine the state and performance of the aircraft. Equation (IV-32) defines the altitude rate of change as the negative of the Earth Z-axis velocity. Integrating (IV-32) gives the altitude, h, while Equation (IV-33) defines the flight path angle, $\gamma$. Equations

(IV-34) and (IV-35) define the normal load factors at the center of gravity and at the accelerometer. The side load factors at the center of gravity and at the accelerometer are described by Eqs (IV-36) and (IV-37) respectively. The distance from the accelerometer to the center of gravity is 9.96 feet along the body X-axis and 0.289 feet along the body Z-axis. Because controllers are designed using dynamic equations linearized at a specified constant velocity, equation (IV-38) calculates the thrust required to zero out acceleration along the body X-axis.

$$\dot{h} = -\dot{Z}_e \qquad \text{(IV-32)}$$

$$\gamma = \arcsin(\dot{h}/V_T) \qquad \text{(IV-33)}$$

$$N_{Zcg} = (\bar{q}S/mg)(C_{D_{TOTAL}}\sin\alpha + C_{L_{TOTAL}}\cos\alpha \qquad \text{(IV-34)}$$

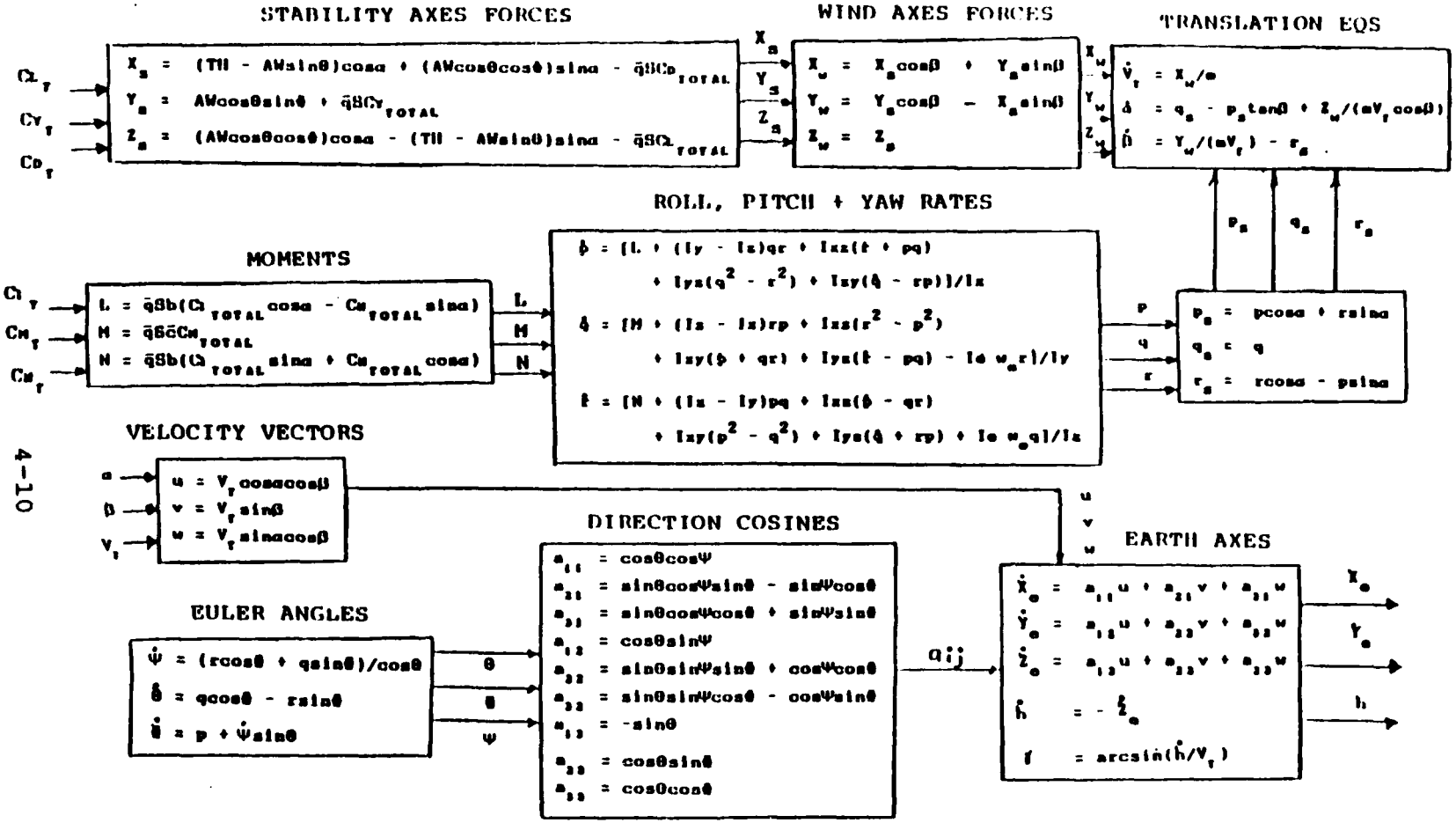
$$N_z = N_{Zcg} + [(9.96 + \bar{c}CG)(\dot{q} - pr) + 0.289(p^2 + q^2)]/g \qquad \text{(IV-35)}$$

$$A_{ycg} = (Y_s/mg) \qquad \text{(IV-36)}$$

$$A_y = A_{ycg} + [(9.96 + \bar{c}CG)(\dot{r} + pq) + 0.289(qr - \dot{p})]/g \qquad \text{(IV-37)}$$

$$TH = AW\sin\theta + (C_{D_{TOTAL}} - AW\cos\theta\ \sin\alpha)/\cos\alpha \qquad \text{(IV-38)}$$

The overall relationship among all the of defined equations is shown in the simulation flow diagram of Figure 4.3.




4.3 Equations of Motion Simulation Flow Diagram

Atmospheric Model

The atmospheric model defines the density, sonic velocity, Mach number, dynamic pressure, static pressure, and total pressure. For the purposes of this simulation, the model contains two regions of the atmosphere, the troposphere and the stratosphere. The troposphere is defined as the portion of the atmosphere starting at the Earth's surface and extending to an altitude of 36,089.24 feet. The stratosphere is the portion of the atmosphere extending above the troposphere. The equations in this section are taken from Reference (21).

Air Density. Air density is described by the gas laws of Boyle and Charles, which detail the relationship among temperature, pressure, and density of a gas. For the purposes of the simulation, assumptions are made that temperature is a linear function of altitude in the troposphere, and a constant over the region of the stratosphere to 50,000 feet. This provides the following relationship between air density at sea level, and the air density at any arbitrary altitude, $\rho$.

$$\rho = \rho_O(1.0 - 0.00000688 * h)^{4.256} \qquad \text{(IV-39)}$$

where $\rho_O$ is given as 0.002378 slug/cubic ft.

Equation (IV-39) is used to form a set of tabulated values, stored in the computer's function generator, and accessed for calculation during simulation run-time.

Sonic Velocity. The sonic velocity is necessary for calculating the Mach number used in the aerodynamic data base. The sonic velocity is a function of altitude, except in the stratosphere region where it is assumed constant due to the assumed constant temperature.

For $h < 36089.24$ ft:

$$a = 1116.45(1.0 - 0.00000688 * h)^{\frac{1}{2}} \quad \text{ft/sec} \qquad \text{(IV-40)}$$

For $h \geq 36089.24$ ft:

$$a = 968.088 \quad \text{ft/sec}$$

Mach Number. The Mach number is the speed of the aircraft compared to the speed of sound. Mach number is an independent variable in the aerodynamic data base.

$$MACH = V_T/a \qquad \text{(IV-41)}$$

Dynamic Pressure. The dynamic pressure combines the effects of velocity and ambient air pressure on the aircraft. The undimensionalized aerodynamic coefficients are multiplied by dynamic pressure to obtain forces.

$$\bar{q} = 0.5\ \rho V_T^2 \qquad \text{lb/ft}^2 \qquad \text{(IV-42)}$$

Static Pressure. The static pressure is calculated according to the atmospheric region of operation. In the troposphere:

$$p_S = p_O(1.0 - 0.00000688 * h)^{5.2561} \quad \text{lb/ft}^2 \qquad \text{(IV-43)}$$

where $p_O = 2116.216$ lb/ft$^2$ is the sea level static pressure.

In the stratosphere:

$$p_S = 472.68 * \exp[-4.80634 \times 10^{-5} * (h - 36089.24)] \quad lb/ft^2 \quad (IV\text{-}44)$$

Total Pressure. Total pressure, used primarily in gain scheduling of the flight control system, is calculated according to the aircraft subsonic or supersonic flight regime.

For MACH 1.0

$$q_C = p_S * [(1.0 + 0.2*MACH^2)^{3.5} - 1.0] \quad lb/ft^2 \quad (IV\text{-}45)$$

For MACH > 1.0

$$q_C = p_S\{166.921[MACH^7/(7*MACH^2 - 1.0)^{2.5}] - 1.0\}$$

in units of $lb/ft^2$ (IV-46)

Aerodynamic Coefficient Equations

Aerodynamic coefficients are used to calculate the magnitude of forces and moments on the aircraft. Each coefficient, when multiplied by dynamic pressure, gives the contribution of its corresponding aircraft component to the overall lift, drag, or moment of the entire airframe. In this simulation, the aerodynamic data base consists of coefficient values determined through wind tunnel testing.

The coefficients, generally depicted as C or R, are used in the following equations taken from reference (21).

Lift. The overall aircraft lift is determined from:

$$C_L = (C_L + \Delta C_{L_{FLEX}})\Big|_{\delta_H = 0 \text{ deg}} + R_{L_{\delta_H}}(C_L\Big|_{\delta_H = x \text{ deg}} - C_L\Big|_{\delta_H = 0 \text{ deg}}) + (R_{l_q} C_{L_q} q_s - R_{l_q} C_{L_{\dot{\alpha}}})(c/2V_T) + \Delta C_{L_{FT}} \qquad \text{(IV-50)}$$

The coefficient of lift ($C_L$) is evaluated at the current flight condition ($\delta_H$ = x deg) and at the condition with zero symmetrical horizontal tail deflection ($\delta_H$ = 0 deg). As shown by the subscript notation in the previous lift equation.

Side Force.

$$C_{Y_{TOTAL}} = R_{Y_\beta} C_Y + R_{l_{\delta_{FA}}} C_{Y_{\delta_{FA}}} \delta_{FA} + C_{Y_{\delta_{HA_{FLEX}}}} \delta_{HA} + R_{Y_{\delta_r}} C_{Y_{\delta_r}} \delta_r + (C_{Y_p} p_s + R_{Y_{rVT}} C_{Y_r} rs)(b/2V_T) \qquad \text{(IV-51)}$$

where

$$C_{Y_{\delta_{HA_{FLEX}}}} = 4(R_{l_{\delta_a}} C_{Y_{\delta_a}} - R_{l_{\delta_{FA}}} C_{Y_{\delta_{FA}}})$$

$$C_{Y_{\delta_a}} = C_{Y_{\delta_{HA}}} + 0.25(C_{Y_{\delta_{HA}}})$$

Drag.

$$C_{D_{TOTAL}} = C_D + \Delta C_D \qquad \text{(IV-52)}$$

Throughout this thesis, the fixed drag increment ($\Delta C_D$) is set to zero. The incremental $\Delta C_D$ can be used for adding

arbitrary amounts of drag, simulating additional external stores, for example.

Rolling Moment.

$$C_{l_{TOTAL}} = R_{l_\beta} C_l + R_{l_{\delta_{FA}}} C_{l_{\delta_{FA}}} \delta FA + C_{l_{\delta_{HA_{FLEX}}}} \delta HA + R_{l_{\delta_r}} C_{l_{\delta_r}} \delta r + (R_{l_p} C_{l_p} p_s + C_{l_r} r_s)(b/2V_T) + (CG - 0.35) C_{Y_{TOTAL}} (\bar{c} \sin\alpha)/b \qquad \text{(IV-53)}$$

where

$$C_{l_{\delta_{HA_{FLEX}}}} = 0.25(R_{l_{\delta_a}} C_{l_{\delta_a}} - R_{l_{\delta_{FA}}} C_{l_{\delta_{FA}}})$$

$$C_{l_{\delta_a}} = C_{l_{\delta_{FA}}} + 0.25\, C_{l_{\delta_{FA}}}$$

$$CG = (X_{ac} - X_{cg})/c$$

For the purpose of this thesis, the nominal center of gravity component (CG) is 0.35. Equation (IV-53) is included in the simulation program to allow simulation at CG values other than nominal.

Pitching Moment.

$$C_{M_{TOTAL}} = C_{M_{0.35\bar{c}}} + (CG - 0.35)(C_{L_{TOTAL}} \cos\alpha + C_{D_{TOTAL}} \sin\alpha) \qquad \text{(IV-54)}$$

where

$$C_{M_{0.35\bar{c}}} = (C_M - \Delta C_{FLEX})\big|_{\delta_H = 0 \text{ deg}} + {}_M\delta_H (C_M\big|_{\delta_H = x \text{ deg}} - C_M\big|_{\delta_H = 0 \text{ deg}}) + (R_{M_q} C_{M_q} q_s + R_{M_q} C_{M_{\dot{\alpha}}} \dot{\alpha})(\bar{c}/2V_T) + \Delta C_{M_{FT}} + \delta C_M$$

Increment $\delta C_M$, set equal to zero in this thesis, is a component available for the addition of pitching moment if other than the nominal CG=0.35 is used. In this thesis, both these terms are zero.

Yawing Moment.

$$C_{n_{TOTAL}} = C_{n_{0.35\bar{c}}} + (CG - 0.35) C_{Y_{TOTAL}} \bar{c} (\cos\alpha)/b \qquad \text{(IV-55)}$$

where

$$C_{n_{0.35\bar{c}}} = R_{N_\beta} C_n + R_{l_{\delta_{FA}}} C_{n_{\delta_{FA}}} \delta_{FA} + C_{n_{\delta_{HA_{FLEX}}}} \delta_{HA} + R_{N_{\delta r}} C_{n_{\delta r}} \delta r + (C_{n_p} p_s + R_{N_{rVT}} C_{n_r} R_s)(b/2V_T)$$

$$C_{n_{\delta_{HA_{FLEX}}}} = 4(R_{l_{\delta_a}} C_{n_{\delta_a}} - R_{l_{\delta_{FA}}} C_{n_{\delta_{FA}}})$$

$$C_{n_{\delta_a}} = C_{n_{\delta_{FA}}} + 0.25(C_{n_{\delta HA}})$$

As with the rolling and pitching moments, this thesis assumes CG = 0.35.

Aerodynamic Data Format

The aerodynamic coefficients (the C-variables) are dependent upon dynamic flight conditions. Through wind

tunnel experimentation (24) the values for these coefficients are determined and tabulated. As the simulation proceeds, the function generator "looks up" the appropriate coefficient value for current flight conditions. When flight conditions fall between tabulated data points, the computer's function generator uses linear interpolation to approximate an appropriate value.

The data table is arranged in groups, as follows.

Group I: Longitudinal Data. The lift, drag, and pitching moment coefficients are in this group of 6300 entries have the following format:

$C_L = C_L(\alpha, \text{MACH}, \delta_H, \delta_{LEF})$

$C_D = C_D(\alpha, \text{MACH}, \delta_H, \delta_{LEF})$

$C_M = C_M(\alpha, \text{MACH}, \delta_H, \delta_{LEF})$

The independent variable ranges and breakpoint values are

$\alpha$ = -4 to +34 degrees in 2 degree increment

MACH = 0.2, 0.6, 0.8, 0.9, 1.2, 1.6, 2.0

$\delta_H$ = -25, -10, 0, 10, 25 degrees

$\delta_{LEF}$ = 0, 15, 25 degrees

where

$\alpha$ is angle of attack,

MACH is Mach number,

$\delta_H$ is symmetrical horizontal tail deflection, and

$\delta_{LEF}$ is leading-edge flap deflection.

Group II: Lateral Directional Data. The side force, rolling moment, and yawing moment coefficients are in this group of 3276 entries and they have the following format:

$C_Y = C_Y(|\beta|, \text{MACH}, \alpha, \delta_{LEF})$

$C_l = C_l(|\beta|, \text{MACH}, \alpha, \delta_{LEF})$

$C_n = C_n(|\beta|, \text{MACH}, \alpha, \delta_{LEF})$

The independent variable ranges and breakpoint values are

$|\beta|$ = 0, 1, 2, 3, 4, 6, 8, 10, 12, 14, 16, 18, 20 deg

MACH = 0.2, 0.6, 0.8, 0.9, 1.2, 1.6, 2.0

$\alpha$ = 0, 10, 20, 30 degrees

$\delta_{LEF}$ = 0, 15, 25 degrees

where

$\beta$ is sideslip angle,

MACH is Mach number,

$\alpha$ is angle of attack, and

$\delta_{LEF}$ is leading-edge flap deflection.

All lateral-directional data are assumed to vary symmetrically with sideslip angle. This is why the absolute value, $|\beta|$, is used as the independent variable. The sign of $C_Y$, $C_l$, and $C_n$ is determined by multiplying the tabulated value by the sign of $\beta$.

Group III: Roll and Yaw Effects. This group contains the linearized control derivatives due to flaperon and differential horizontal tail deflections. There are 1680 values in the following format:

$$C_{l_{\delta_{FA}}} = C_{l_{\delta_{FA}}}(\alpha, \text{MACH}, \delta_{LEF})$$

$$C_{n_{\delta_{FA}}} = C_{n_{\delta_{FA}}}(\alpha, \text{MACH}, \delta_{LEF})$$

$$C_{l_{\delta_{HA}}} = C_{l_{\delta_{HA}}}(\alpha, \text{MACH}, \delta_{LEF})$$

$$C_{n_{\delta_{HA}}} = C_{n_{\delta_{HA}}}(\alpha, \text{MACH}, \delta_{LEF})$$

The independent variable ranges and breakpoint values are

$\alpha$ = -4 to +34 degrees in 2 degree increments

MACH = 0.2, 0.6, 0.8, 0.9, 1.2, 1.6, 2.0

$\delta_{LEF}$ = 0, 15, 25 degrees

Group IV: Side Force Effects and Flight Test Correction. This group of 560 entries contains the side force effects of the flaperons and the differential horizontal tail. This group also includes flight test corrections to the pitching moment and lift coefficients, denoted by $\Delta$. The coefficients are listed sequentially in the table, and have the following format:

$$C_{Y_{\delta_{FA}}} = C_{Y_{\delta_{FA}}}(\alpha, \text{MACH})$$

$$C_{Y_{\delta_{HA}}} = C_{Y_{\delta_{HA}}}(\alpha, \text{MACH})$$

$$\Delta C_{L_{FT}} = \Delta C_{L_{FT}}(\alpha, \text{MACH})$$

$$\Delta C_{M_{FT}} = \Delta C_{M_{FT}}(\alpha, \text{MACH})$$

with the following independent variable ranges and breakpoint values:

$\alpha$ = -4 TO +34 degrees in 2 degree increments

MACH = 0.2, 0.6, 0.8, 0.9, 1.2, 1.6, 2.0

Group V. Rudder Effects. This group of 1260 entries contains the rudder effects in the following format:

The independent variable ranges and breakpoint values are

$$Cy_{\delta_r} = Cy_{\delta_r}(\alpha, \text{MACH}, |\delta_r|)$$

$$Cn_{\delta_r} = Cn_{\delta_r}(\alpha, \text{MACH}, |\delta_r|)$$

$$Cl_{\delta_r} = Cl_{\delta_r}(\alpha, \text{MACH}, |\delta_r|)$$

with the following independent variable ranges and breakpoint values:

$\alpha$ = -4 to +34 degrees in 2 degree increments

MACH = 0.2, 0.6, 0.8, 0.9, 1.2, 1.6, 2.0

$|\delta_r|$ = 10, 20, 30 degrees

where

$\delta_r$ is rudder deflection.

The tabulated coefficients are for positive rudder deflection. For rudder deflections in the negative direction, the assumption is made that the corresponding coefficient is the same. Rudder deflections of less than 10 degrees default to the 10 degree coefficient value.

Group VI: Dynamic Derivatives. This group includes lift and pitching moment derivatives with respect to pitch rate and rate of change of angle of attack; it includes side force, rolling and yawing moment derivatives with

respect to roll and yaw rate. There are 1400 entries in this group, and they are in the following format:

$$C_{L_q} = C_{L_q}(\alpha, \text{MACH})$$

$$C_{M_q} = C_{M_q}(\alpha, \text{MACH})$$

$$C_{L_{\dot{\alpha}}} = C_{L_{\dot{\alpha}}}(\alpha, \text{MACH})$$

$$C_{M_{\dot{\alpha}}} = C_{M_{\dot{\alpha}}}(\alpha, \text{MACH})$$

$$C_{Y_p} = C_{Y_p}(\alpha, \text{MACH})$$

$$C_{l_p} = C_{l_p}(\alpha, \text{MACH})$$

$$C_{n_p} = C_{n_p}(\alpha, \text{MACH})$$

$$C_{Y_r} = C_{Y_r}(\alpha, \text{MACH})$$

$$C_{l_r} = C_{l_r}(\alpha, \text{MACH})$$

$$C_{n_r} = C_{n_r}(\alpha, \text{MACH})$$

The independent variable ranges and breakpoint values are

$\alpha$ = -4 to +34 degrees in 2 degree increments

MACH = 0.2, 0.6, 0.8, 0.9, 1.2, 1.6, 2.0

Group VII: Flex/Rigid Ratios. The flexibility corrections, in terms of flex/rigid ratios, due to rotation rates, surface positions, and sideslip angle are in this group of 1140 entries.

$$R_{l_p} = R_{l_p}(\text{MACH}, H_1)$$

$$R_{l_q} = R_{l_q}(MACH, H_1)$$

$$R_{M_q} = R_{M_q}(MACH, H_1)$$

$$R_{Y_{rVT}} = R_{Y_{rVT}}(MACH, H_1)$$

$$R_{n_{rVT}} = R_{n_{rVT}}(MACH, H_1)$$

$$R_{l_{\delta_H}} = R_{l_{\delta_H}}(MACH, H_1)$$

$$R_{M_{\delta_H}} = R_{M_{\delta_H}}(MACH, H_1)$$

$$R_{Y_{\delta_r}} = R_{Y_{\delta_r}}(MACH, H_1)$$

$$R_{l_{\delta_r}} = R_{l_{\delta_r}}(MACH, H_1)$$

$$R_{n_{\delta_r}} = R_{n_{\delta_r}}(MACH, H_1)$$

$$R_{l_{\delta_a}} = R_{l_{\delta_a}}(MACH, H_1)$$

$$R_{l_{\delta_{FA}}} = R_{l_{\delta_{FA}}}(MACH, H_1)$$

$$R_{Y_\beta} = R_{Y_\beta}(MACH, H_2)$$

$$R_{l_\beta} = R_{l_\beta}(MACH, H_2)$$

$$R_{n_\beta} = R_{n_\beta}(MACH, H_2)$$

The independent variable ranges and breakpoint values are

MACH = 0.2 to 2.0 in 0.1 increments

$H_1$ = 0, 10,000, 25,000, 50,000 feet

$H_2$ = 0, 20,000, 40,000, 50,000 feet

where

$H_1$ and $H_2$ are altitude.

Group VIII: Flexibility Increments. The flexibility increments of lift and pitching moment are in this group of 3200 entries. They have the following format:

$$\Delta C_{L_{FLEX}} = \Delta C_{L_{FLEX}}(\alpha, H_1, \delta_{LEF}, MACH)$$

$$\Delta C_{M_{FLEX}} = \Delta C_{M_{FLEX}}(\alpha, H_1, \delta_{LEF}, MACH)$$

The independent variable ranges and breakpoint values are

$\alpha$ = -4 to +34 degrees in 2 degree increments

$H_1$ = 0, 10,000, 25,000, 50,000 feet

$\delta_{LEF}$ = 0, 5, 10, 25 degrees

MACH = 0.2, 0.6, 0.9, 1.2, 1.6

Backward Difference Equation

To implement digitally a flight control system designed in the LaPlace domain, the simple backward difference equation is used. Transfer functions are thereby converted to digital time-domain form for insertion into a FORTRAN digital flight control routine (12:113-6). The first backward difference equation (11:90-4) is

$$Dc(kT) = \frac{dc(t)}{dt}\bigg|_{t=kT} \approx \frac{c(kT) - c[(k-1)T]}{T} \quad \text{(IV-56)}$$

where

k = digital time increments, 1, 2, 3, ...

T is the sampling interval,

D is the derivative operator.

As an example of its use, consider the typical lag transfer function

$$\frac{C(s)}{U(s)} = \frac{10}{s + 10} \qquad \text{(IV-57)}$$

The first step is to rearrage Eq (IV-57) and take the inverse LaPlace Transform.

$$(s + 10)\ C(s) = 10\ U(s)$$

$$(D + 10)\ c(t) = 10\ u(t)$$

$$Dc(t) + 10\ c(t) = 10\ u(t) \qquad \text{(IV-58)}$$

Substituting Dc(t) in Eq (IV-58) with Eq (IV-56) when t=kT to derive the difference equation at the discrete computation points.

$$\frac{c(kT) - c[(k - 1)T]}{T} + 10\ c(kT) = 10\ u(kT)$$

Solving for c(kT) yields

$$c(kT) = \frac{10T\ u(kT) + c[(k - 1)T]}{1 + 10\ T} \qquad \text{(IV-59)}$$

As shown in Eq (IV-59), the output c(kT), is computed from the previous output, the current input, and the interval of computation. To use this method of conversion, the order of the transfer function denominator must be greater than or equal to the order of the numerator. This conversion technique may be extended to higher-order

transfer function, as explained in Reference (11), pages 89-96.

The equations presented in this chapter are used as a basis for modeling the aircraft and its environment on the SIMSTAR computer.

# V. Simulation Implementation

## Overview

The final implementation of SIMTACS-RT is described, including apportionment of various simulation tasks among simulator components. In addition, resolution of some of the issues involved in the condensation of SIMTACS-RT to a single SIMSTAR computer are described.

## SIMTACS-RT General Structure

The overall simulation is implemented as shown in Figure 5.1. The heart of the simulator is the EAI SIMSTAR hybrid computer. Pilot control inputs are fed, through a force stick, to the computer. The computer provides feedback to the pilot, for man-in-the-loop testing via a visual output device, in this case an oscilloscope set up to provide pitch and bank information. Data is recorded for each run on strip chart recorders and in digital data logger files.

## The Hybrid Computer

The simulation tasks for the computer are divided according to the functional and speed capabilities of each major section of the computer. See Figure 5.2.

Real-time calculations, including those supporting the aircraft dynamic model, are performed in the SIMSTAR's analog (continuous-time solution) processor. The analog processor also interfaces the SIMSTAR to external devices

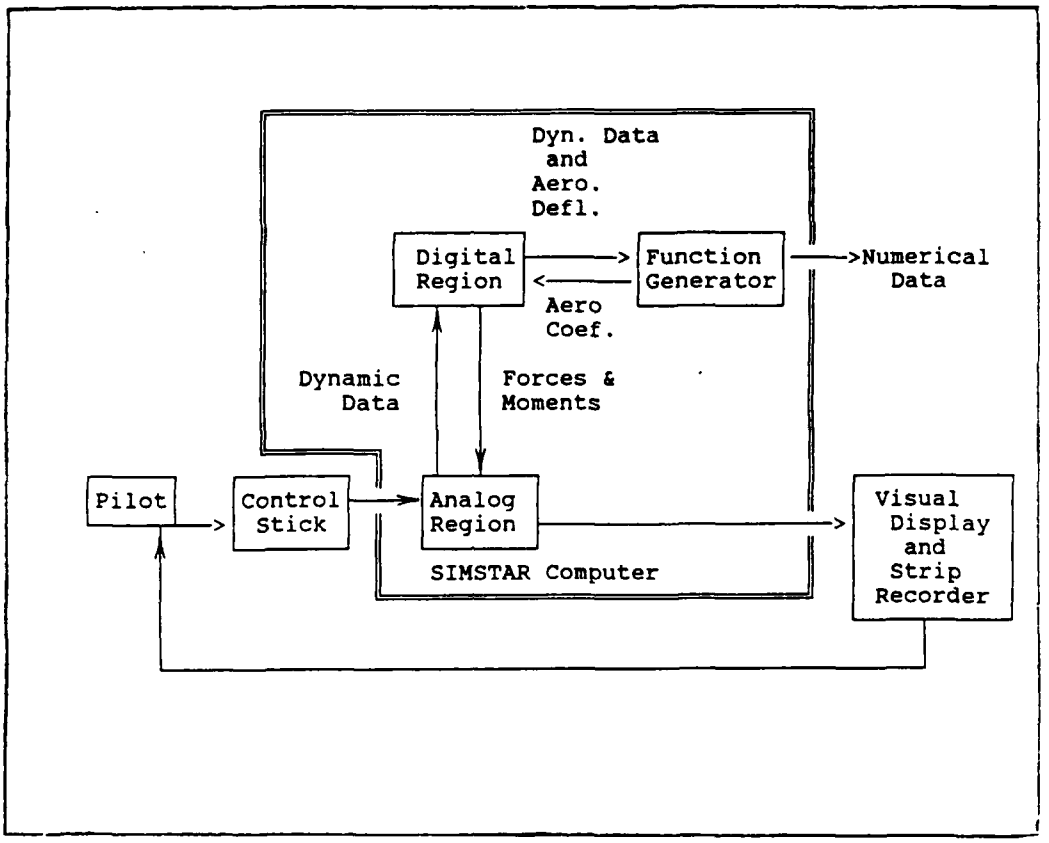



Figure 5.1 The Closed Loop

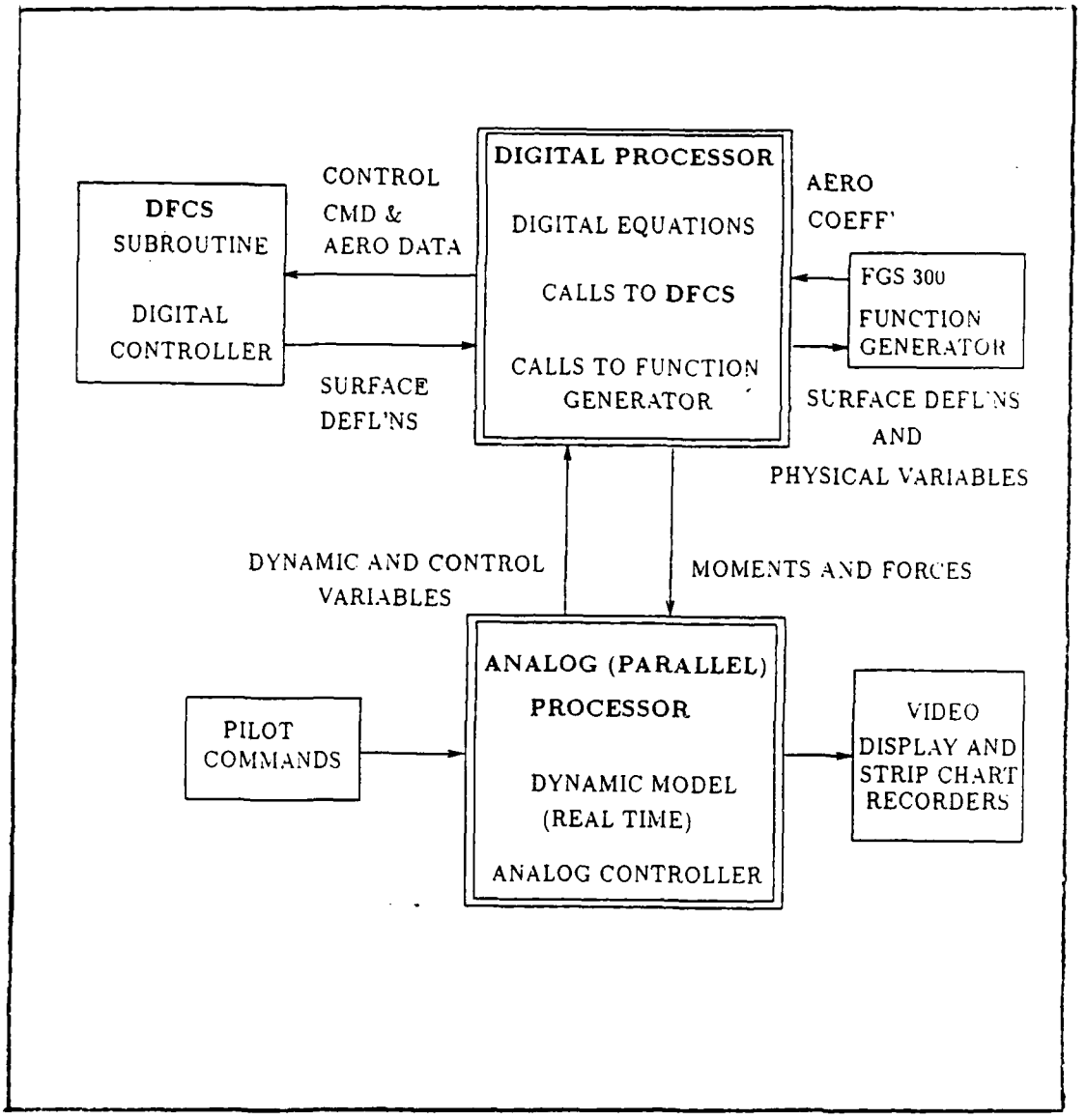



**Figure 5.2 Computer Architecture Utilization**

including the pilot's control stick, the external video display, and the strip chart recorders.

The digital region of the SIMSTAR is used for modeling of the digital flight control system, making calls to the function generator for lookup of aerodynamic coefficient values, and for calculation of forces and moments from those coefficients, using the equations described in Chapter IV. Those forces and moments are transferred to the analog processor for use in the dynamic model.

If all of SIMSTAR's analog components are in use, the digital processor can be used to assist with the dynamic calculations, at a cost of non-continuous operation. Elements of the dynamic model considered least time-critical from the standpoint of studying the aircraft controller dynamics are the best candidates for digital implementation. An example is the integration of vertical velocity to determine altitude.

The FGS 300 function generator, built by EAI, provides the computer's digital region with aerodynamic coefficients computed from experimentally-determined tables (24). The aerodynamic coefficients described in Chapter IV are tabulated and stored in the FGS 300 at the initialization of each simulator run. As the simulation proceeds, the function generator looks up the most appropriate function value for each aerodynamic coefficient based on external conditions provided to it by the

SIMSTAR's digital processor. To enhance precision, the FGS 300 performs high-speed multi-dimensional interpolations.

The external parameters passed to the FGS 300 are angle of attack ($\alpha$), Mach number (M), symmetrical horizontal tail deflection ($\delta_H$), leading edge flap deflection ($\delta_{LEF}$), sideslip angle ($\beta$), rudder deflection ($\delta_r$), and altitude ($H_1$ or $H_2$). The FGS 300 returns all the aerodynamic coefficients described in Chapter IV.

Dynamics Model Routine

The aircraft dynamics model is represented by equations which describe the physical plant characteristics of the airplane. Most of these equations reside in the analog region of the SIMSTAR program, with the exception of a few which are moved to the digital region, as described above.

The SIMSTAR's analog capabilities allow for the simultaneous, real-time computation of interdependent, non-linear differential equations. Mathematical operations such as integration, multiplication, and trigonometric function evaluation are performed by internal components specifically designed for these functions. Connections among the analog components are made automatically by the SIMSTAR through an analog switch matrix.

Control System Routine

The digital flight control system (DFCS) is contained in a FORTRAN subroutine accessed through the digital derivative region of the SIMSTAR code. This subroutine is intended to represent the control law provided by the user/designer. Inputs to the subroutine include pilot input signals and feedback variables required for the flight control law computation. The subroutine's outputs are the control surface deflections.

To use the program, the user designs his flight control system in the s-domain in transfer function form. These transfer functions are coded as time difference equations, as described in Chapter 4. Alternatively, the design can be performed in the z-domain directly. The DFCS subroutine equations must be written to store all calculated values at each sample time. At each succeeding digital time increment, these values are used to calculate updated values.

Control surface actuators have been modeled digitally in the DFCS subroutine. This is to conserve analog components for use in high-speed dynamic calculations, and to avoid digital processing lags. Had the actuators been modeled in the analog processor, one digital computation cycle would be needed to read the DFCS commands and pass them to the analog processor (as actuator input). Another digital computation cycle would have been required to read

the actuator response from the analog processor and feed it to the FGS 300 for table look-up.

An analog controller can be implemented on SIMTACS-RT as well. The program source code is designed in such a way that it allows the direct insertion of code describing the control law, including feedback loops and gain schedules. The source code includes a verified controller which is ready to run. It is easily modified to study the effect of changes in gain schedules or actuators dynamics.

One current disadvantage of running the analog controller simulation is the necessary limitation to one mode of operation, longitudinal or lateral. The limitation is caused by the limited number of analog computing blocks within the SIMSTAR. This limitation can be eliminated by slaving the second SIMSTAR in analog mode, thus doubling the analog computation capability.

For more explicit information on coding an aircraft system design, see Appendix B, the SIMTACS-RT User's Manual. Mark Kassan (12:113-22) also provides an excellent general procedure for implementing a flight control law on SIMSTAR.

SIMTACS-RT Runtime Structure

The first part of the SIMSTAR program code, called INITIAL, sets the output values of the DFCS difference equations initially to zero. The INITIAL region also calculates coefficients of the difference equations, which

are based on the sampling period of the control system. The sampling time used during the simulation is determined by the parameters PERIOD, CINT, BET, and NCYCLE. These parameters are set by the user at the beginning of the simulation run.

Briefly, the four parameters work as follows. Variable CINT is the communication interval used for data logging. The value of CINT determines the speed at which an entire set of required data may be read from the parallel region and passed along to the digital (derivative) region of the computer. Variable PERIOD sets the computation speed for digital calculations. For example, setting PERIOD equal to a value of 0.01 would update the real-time simulation in the digital region at 100 Hertz.

Variable BET determines the simulation speed relative to real time. Setting BET equal to 1.0 causes the simulation to run in real-time. Doubling BET scales the simulation to run at half-time, and so on. The parameter NCYCLE determines the frequency at which the DFCS operates. Setting NCYCLE at 4 (only integer values are allowed) will invoke the DFCS subroutine every fourth digital computation period. This arrangement allows the user to run SIMTACS-RT at the fastest speed at which the SIMSTAR is capable of running, yet run the DFCS at a rate required by the DFCS. The advantage is that the dynamic and kinematic equations and aerodynamic coefficients are

updated as quickly as possible for maximum fidelity. Should a digitally-driven video display be added to SIMTACS-RT, a quicker digital computation rate can also reduce flicker and time lag which degrade the overall man-in-the-loop simulation fidelity.

Running SIMTACS-RT

While detailed instructions for coding and running SIMTACS-RT are provided in the appendices, the basic procedure is as follows.

The flight control designer programs his digital control law in FORTRAN 77 code. The argument list of his code should be arranged to match the argument list in the SIMTACS-RT DFCS subroutine call. His flight control code, in subroutine form, is appended to the end of the SIMTACS-RT code using the SIMSTAR's text editor. The entire program is compiled as a unit.

Once the program is compiled, it is loaded into the SIMSTAR runtime executive. The user sets his timing from the keyboard using the BET, PERIOD, CINT, and NCYCLE parameters described. To use SIMTACS-RT in a man-in-the-loop configuration, the video display and control stick driver must be online. The SIMTACS-RT simulation is ready for flight. Data is collected on the strip chart recorder and digital data-logger.

New Issues

The simulator is transformed into its current configuration, SIMTACS-RT, by resolving four main issues;

1. Single-machine Operation,
2. FGS 300 Integration
3. Real-time Operation
4. Digital/Analog Controller Operation

Condensing the simulator for implementation on one SIMSTAR computer is the first issue to be resolved. Computer code which interfaces the two computers is rewritten to re-route data to different processors within one computer. In most cases, this consists of additional analog-to-digital data transfer code. This is required because the previous version of SIMTACS used another SIMSTAR to perform function generation and to calculate forces and moments from aerodynamic coefficients.

The equations which use the aerodynamic coefficients for force and moment generation are also added to the derivative region of the SIMTACS-RT code (see Figure 5.3 for SIMTACS-RT program structure). Additionally, the DFCS subroutine is placed within a PROCEDURAL block. PROCEDURAL blocks are used to ensure portions of program code remain intact during the SIMSTAR compilation process (18:20).

The integration of the FGS 300 is the key to making SIMTACS-RT a one-machine simulator. Documentation

```
(STARTRAN Processing Directives)
   PROGRAM name
     (Global System Declarations)
     (Program Initialization)
        INITIAL PROGRAM REGION
          (Computer Runtime Initialization)
        END OF INITIAL
        DYNAMIC PROGRAM REGION
          (Lowest Priority Runtime Procedures)
            DERIVATIVE PROGRAM REGION
              (Digital Processor Code)
            END OF DERIVATIVE REGION
        END OF DYNAMIC REGION
        TERMINAL PROGRAM REGION
          (Exit Program or Examine Results of Run)
        END OF TERMINAL REGION
        END OF PROGRAM
          (Subroutines for Digital/Analog Interface)
          (DFCS Subroutine)
          (FGS 300 Loading and Runtime Subroutines)
```

Figure 5.3 Typical SIMSTAR Program Structure

provided by EAI (25) was oriented toward use of the FGS 300 within a FORTRAN run in batch mode. A great deal of trial-and-error was required to coordinate the operation of the FGS 300 in real time. Another problem, which presented itself after the FGS 300 was operational, involved the use of memory by SIMSTAR. SIMSTAR uses active memory to store function values it loads into the FGS 300. Because the total number of stored data points exceeded 20,000 values, the FGS 300 loading process leads to memory conflicts. This problem is solved through the use of extended memory procedures (18:139). A procedure for integrating the FGS 300 into a SIMSTAR program is detailed in Appendix A.

Once again, additional interface code is written to ensure the proper physical variables are transferred to the FGS 300. The function values returned by the processor also must be made available to the digital processor for use in the generation of forces and moments.

Real-time operation is made possible mainly due to the integration of the FGS 300 into SIMTACS-RT. In the final configuration, SIMSTAR's digital region is able run at a computation cycle time of 7 milliseconds. The FGS 300 has no problem accomodating this computation rate, as it performs table lookups and interpolations at a cycle time of approximately 15 microseconds per function computation, depending on the number of independent

variables used. One powerful feature of this configuration is the ability to run the DFCS at the required rate of speed for an actual controller while running the remaining simulation at the fastest speed possible. For example, SIMTACS-RT can be run with a computational cycle speed of 8 milliseconds, with the DFCS updated every third computational cycle. This means the DFCS is running at a computational cycle of 24 milliseconds, or approximately 40 Hertz, a typical update rate for a digital control system.

SIMTACS-RT has available a collection of programs and supporting code for the testing of aircraft control laws. The following programs are available.

1. FEP--A program written to simulate longitudinal control laws on the F-16. Code is in place to provide ideal inputs, steps and pulses, of variable length. FEP contains an analog and a digital control law, which can be run sequentially for comparison.

2. MANLO--A program similar to FEP which allows real-time man-in-the-loop operation. The force stick provides pilot input and video display output is available on the oscilloscope.

3. TABLE--A program written to load and "dry-run" the FGS 300. This program is useful for testing the subroutines which load and access the FGS 300, and to

investigate the linearity of aerodynamic coefficients at a point in the flight envelope.

4. BIG--This program provides simulation capablility of the coupled longitudinal and lateral aircraft dynamics and control law. Some interface issues remain to be resolved, and BIG is not providing valid data at this time.

Testing and validation of SIMTACS-RT is performed by comparing strip chart data from simulation runs to outputs already validated in the previous version of SIMTACS by Mark Kassan (12:65-106). The control law is inserted into the simulation code, and each loop closed individually. Strip chart data is shown in Figures 5.4-5.9. Plots from trimmed flight and a one-G pullup for both analog and digital control laws is provided.

The data provided by SIMTACS-RT very closely matches the data produced and validated by Mark Kassan. Reproductions of his data are included to assist in the comparison. All test data for Mach 0.9 and 20,000 feet.

Summary

SIMTACS-RT is designed to make optimal use of the unique characteristics of the SIMSTAR computer. SIMTACS-RT takes advantage of the continuous simulation capability of SIMSTAR's hybrid region, and the ability of the SIMSTAR's digital region to call a subroutine implementing a digital flight control system. In addition, the digital

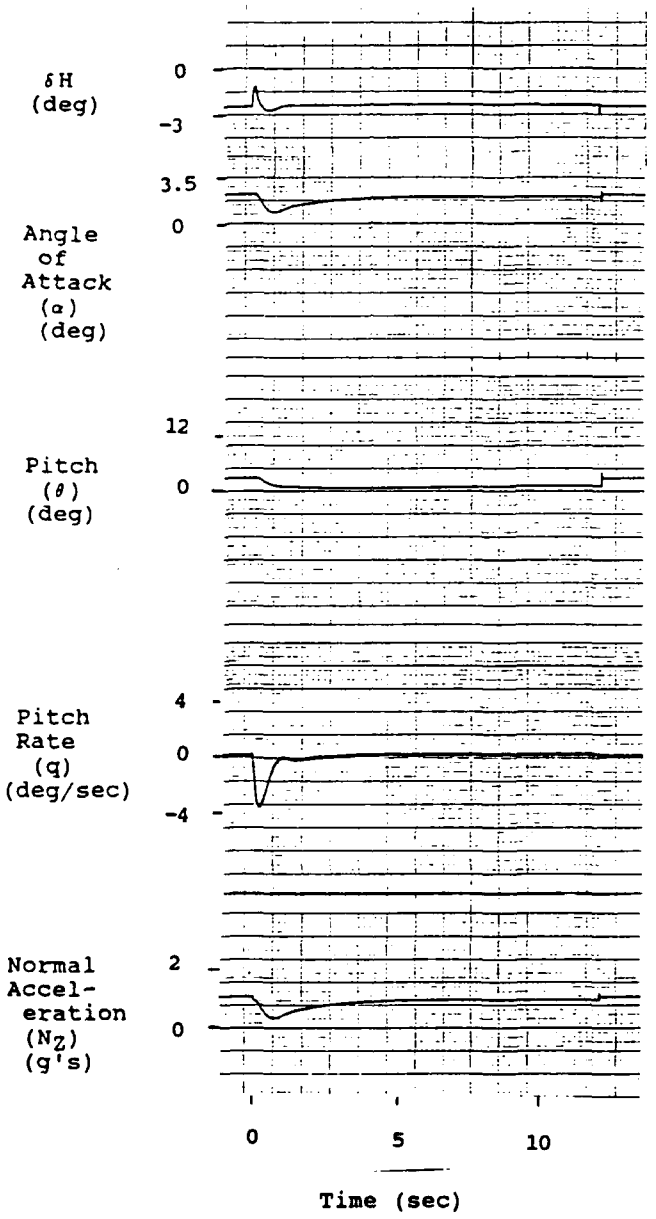



Figure 5.4 Trimmed Flight, Analog Control Law

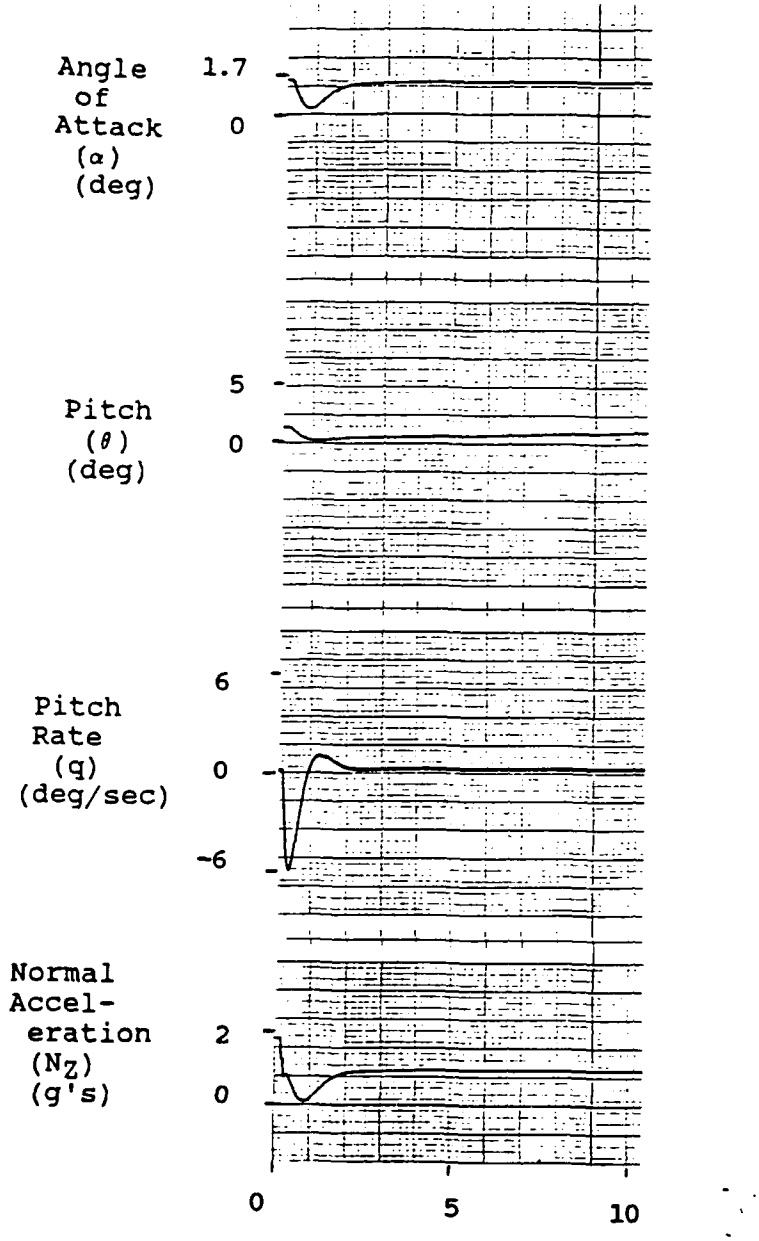



Figure 5.5 Trimmed Flight, Digital Control Law

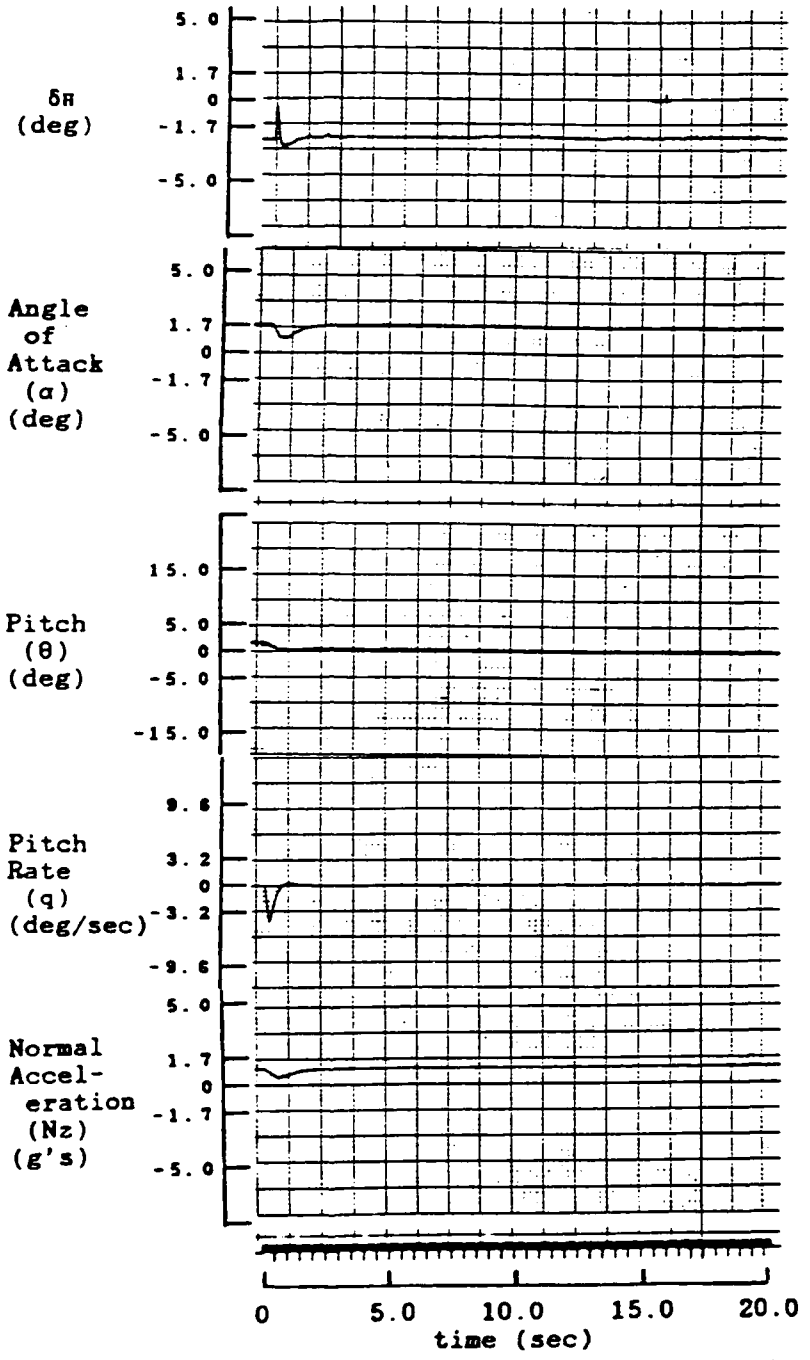



Figure 5.6 Previously Validated Trimmed Flight

(From 12:86-90)

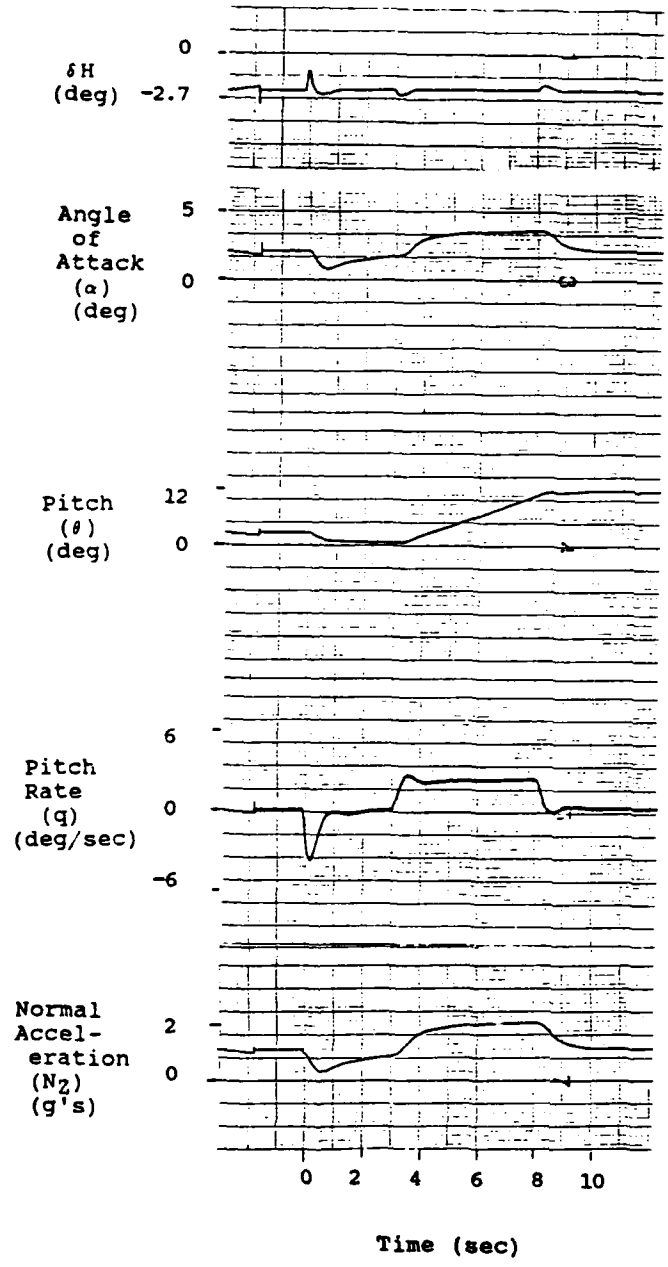



Figure 5.7 1-G Pullup, Analog Control Law

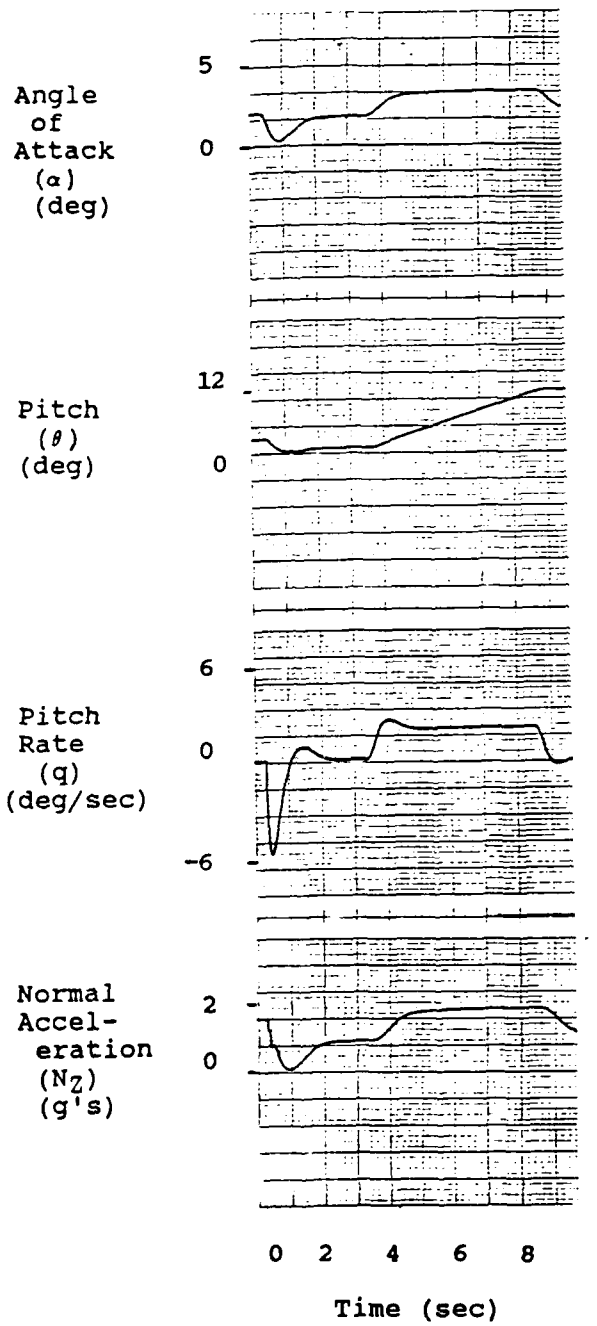



Figure 5.8 1-G Pullup, Digital Control Law

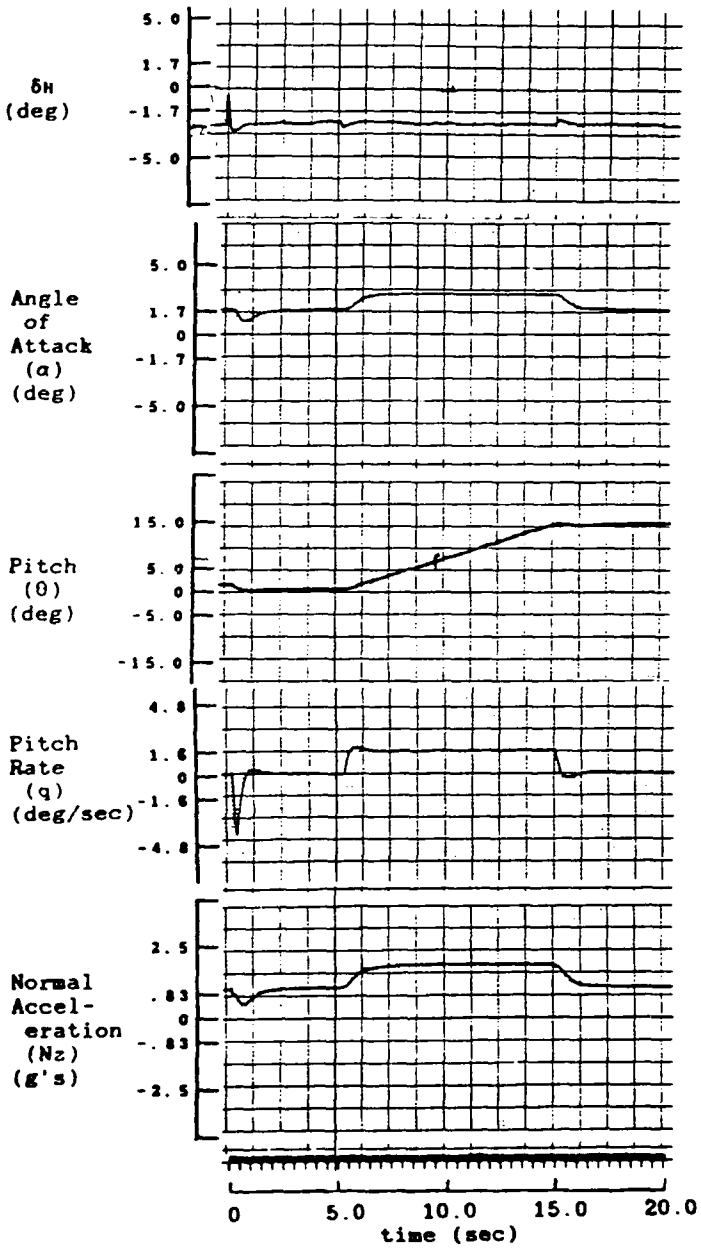



Figure 5.9 Previously Validated 1-G Pullup

(From 12:81-93)

region uses the tremendous speed and flexibility of the FGS 300 to provide the most accurate and rapid coefficient calculation possible.

# VI. Conclusions and Recommendations

## Overview

The real-time simulator SIMTACS-RT has great potential for research applications and as a controller design, analysis, and research tool. It meets criteria for high fidelity, and is capable of true man-in-the-loop operation. Additionally, its library of ready-to-run programs and joint digital/analog capability makes it a valuable tool for use in comparing control law designs.

Further recommendations for improvements to SIMTACS-RT include physical upgrades to the cockpit environment, and acquisition of aerodynamic data libraries for additional F-16 aircraft configurations. Additionally, aerodynamic data for other aircraft should be obtained. The SIMTACS-RT should also be modified for user-friendliness, particularly by improving the units consistency within the program code and aerodynamic data files.

## Conclusions

SIMTACS-RT has excellent potential for applications in research and controller design. The two most important characteristics of SIMTACS-RT are its true high fidelity dynamics and its real-time, man-in-the-loop simulation capability.

What separates SIMTACS-RT from standard CAD simulation is the use of a true non-linear, time varying dynamic

model, in conjunction with empirically-derived aerodynamic data. As described earlier, the simulation capabilities of CAD programs depends on linearizations of the true system dynamics. SIMTACS-RT provides an engineer the ability to expeditiously evaluate his controller design, gaining insight into real-world performance, without the work-intensive procedures necessary with a typical high fidelity simulation facility. Facilities such as those at Wright-Patterson AFB are already in high demand. Promising designs can be selected early as candidates for the more elaborate simulations.

The addition of the FGS 300 function generator, along with the condensation of SIMTACS-RT to a single SIMSTAR computer, has made possible real-time, man-in-the-loop simulation. While consideration of the pilot's interaction with the flight control system and aircraft had been limited to linear approximations, as discussed in Chapters I and II, preliminary real-time data can now be collected easily. This represents a new capability for research at AFIT.

To assist in the study of controller designs, there are several ready-to-run programs and routines available, as enumerated in the previous chapter. These programs simplify the testing process for the engineer. He has at his disposal programs to perform the standard control law testing procedures, including system inputs such as pulses

and impulses. Also included within runtime routines are initialization data for flight conditions at 20,000 feet at velocities of Mach 0.8 and Mach 0.9.

The programs FEP and MANLO, discussed earlier, have the capability to test both digital and analog control laws. The programs are written modularly as well, so that the capability exists for "fly-offs" among several designs of interest.

Recommendations

With a high-fidelity dynamic model implemented on SIMTACS-RT, and real-time man-in-the-loop capability proven, the prospects for improvements to SIMTACS-RT look promising. The recommendations fall into two categories, improvements to SIMTACS-RT, and new uses of SIMTACS-RT.

First, there remains improvement of simulation realism such as the cockpit environment. The oscilloscope presentation used as a visual display needs to be replaced with a more realistic presentation. AFIT has recently acquired a Sun 3/160 workstation with a video accelerator board for use with SIMTACS-RT. Integration of a realistic video display should be a high priority. AFIT itself has the ability to develop the video software, and also has several graphic display programs available. Additionally, the Flight Dynamics Laboratory is performing extensive research into simulation video displays, and promising concepts are available for integration into SIMTACS-RT.

Care must be excercised in selection of a video display routine for use with SIMTACS-RT. Video display code which is too elaborate may run slowly. This resulting display time lag may negate the advantage provided by SIMTACS-RT's real-time analog processor. This must be considered in the integration of a new video system into SIMTACS-RT.

An actual aircraft control stick and throttle quadrant is needed to improve simulation realism. The author's flying experience indicates that pilots prefer to adjust their own trim and power settings as they fly. The current necessity of making those settings through computer keyboard inputs denies the pilot that added increment of realism. AFIT has been exploring acquisition possibilities with regard to this equipment. Integration of the stick and throttle quadrant into SIMTACS-RT through external analog inputs is straightforward.

The aerodynamic tables provide data for flight conditions from sea level to 50,000 feet, from Mach 0.2 to Mach 2. Early validation of the simulator throughout this flight envelope is required. Additionally, the aerodynamic data tables allow the use of leading edge flaps, ignored in this study. The validation of the leading edge flaps' use in simulation requires additional code in the aircraft control law, comparison of simulator output to test data.

Data for other F-16 aircraft configurations, such as extended landing gear and various armament combinations should be added to the aerodynamic library and validated. Additional cockpit components, such as landing gear handle, flap lever, and other equipment should be interfaced to SIMTACS-RT through the external analog ports, and used as switches to change the aircraft flight configuration.

A great deal of effort was expended in assuring the correct use of engineering units, i.e. degrees and radians, feet/sec$^2$ and G-load. Indeed one problem with the integration of the FGS 300 is the fact that the aerodynamic tables are tabulated according to degrees, but the function values themselves are in radians$^{-1}$. Keeping track of the units complicates the use of SIMTACS-RT by an engineer not intimately familiar with the details of SIMSTAR hardware and software. Programming the code in only one system of units, or inserting a "menu" to automatically select units can improve SIMTACS-RT's "user friendliness."

The final recommendation is to make maximum use of SIMTACS-RT. As discussed in earlier chapters, SIMTACS-RT provides AFIT a capability to perform a high-fidelity, real-time simulation. This is a vast improvement over batch-mode linearized simulations. The SIMTACS-RT

simulations can even be used to select promising designs for simulation on more elaborate simulators.

As SIMTACS-RT is used, simulation procedures can become more refined, and new uses are likely to be discovered. The SIMTACS-RT should not be limited to simple aircraft controller simulations. There are possible applications in robotics, telepresence, and cockpit human factors engineering studies to name a few. The SIMSTAR computer, with its ability to provide continuous nonlinear solutions at any desired speed, is a valuable asset, and SIMTACS-RT is a stepping stone toward development of its full potential.

Summary

The SIMTACS-RT holds great promise as a tool for the study of simulation, control, and man-in-the-loop issues. New capabilities have been added to SIMTACS-RT which should be explored and improved upon. It should be used for research and academic work at AFIT, and additional capability should be added.

# Appendix A: Using the FGS 300 Function Generator

## Within a SIMSTAR Program

The documentation supplied by EAI on the use of the FGS 300 assumes the function generator is used as a call from an ordinary FORTRAN program. As a result, some required procedures are not documented. The following explains modifications of EAI's published procedures necessary to use the FGS 300 in a SIMSTAR hybrid program.

The FGS 300 Function Generator. The function generator is able to perform table lookup of tabulated functions. Up to 64 functions may be defined. These functions may be dependent on a maximum of 32 independent variables. The functions may be divided into a maximum of eight function groups. This provides the capability to generate functions at different rates. They may be segregated into the function groups in any desired combination, as long as the combined maximum of 64 dependent functions and 32 independent variables is not exceeded.

The independent variable of a function is defined in a set of breakpoints. These breakpoints specify the values of the independent variable at which dependent function values are known. For example, if z = f(x), then throughout the range of the independent variable x, there

is a set of values of x for which the values of z are known. These values of x are the breakpoints. The corresponding values of z are the tabulated data points. This concept is expanded to functions of multiple variables. When the breakpoints of each independent variable, along with the data points of the dependent variable are fed into the function generator, it is able to perform rapid table lookups rapidly for any value of independent variable within the user-specified range. If the independent value used for the lookup lies between breakpoints, linear interpolation is used to provide a reasonable estimate of function value. The function generator also has the capability to perform linear extrapolation for values of the independent variable outside the defined range.

The SIMSTAR accesses the function generator through FORTRAN subroutines. There are two user-written subroutines required, PREINIT and FGEXEC.

Subroutine PREINIT. PREINIT "feeds" into the function generator the breakpoints of the independent variables along with the corresponding dependent function values.

This subroutine also correlates variable names used by the programmer to firmware-defined function and variable names within the FGS 300. The FGS 300 internally

uses the name FGX(n) for independent variables, and FGF(n) for dependent variables.

The routine PREINIT need not be called explicitly. SIMSTAR looks for a subroutine named PREINIT every time a SIMSTAR program is loaded. If for some reason, a name other than PREINIT is used for this subroutine, an explicit call is required in the PRE-INITIAL portion of the SIMSTAR program.

Refer to Example A.1 for the structure of a sample subroutine called PREINIT. The first line executed in PREINIT is an INCLUDE statment. This INCLUDE call performs memory allocations for the function generator, and allows the compiler to find the subroutines called later. The pathname to the file PFGSIN must be given. The next set of commands dimensions arrays that are used by the FGS 300.

This example is an adaptation of the SIMTACS-RT program which uses the FGS 300 to look up aerodynamic values. The variable names used throughout this example are used in the SIMTACS-RT code. AOA refers to angle of attack, MACH and MII refer to the standard velocity measurement relative to the speed of sound. CL refers to the basic coefficient of lift, and CM refers to basic pitching moment. RMDH and RYDR refer to a flex/rigid ratios. H1 refers to altitude.

There are 20 values of AOA used as breakpoints, 7 breakpoints for MACH, etc. There are 20 x 7 or 140 values of the first two functions, and 19 x 4 or 76 values of the last two functions.

If any function data is in an external table, as all of it is in this example, there must exist an OPEN command to access the data. The data is then read in sequentially. See Example A.2 for data structure.

Notice how the subroutine reads in the table values. It reads in all the values for MACH of 0.2 in order, beginning with AOA of -4.0 degrees and proceeding to 34.0 degrees in 2.0 degree increments. Next are the twenty data points corresponding to MACH of 0.6, etc. Following the CL data points are 140 CM data points in the same pattern.

The RMDH data is similarly arranged. There are 19 data points for H1 of 0, beginning with MII of 0.2, and proceeding to MII of 2.0 in increments of 0.1 units. RYDR is organized in the same manner.

The PINCL call in PREINIT is used for function generator allocation purposes. Unless there are multiple users of the function generator, the PINCL subroutine will not affect operations. If there are more than one users calling the FGS 300, PINCL will notify the latter ones that the FGS 300 is already in use and unavailable.

The PDEFAS and PDEFAT calls are used next to define the breakpoint structure for the FGS 300. A good technique is to group all PDEFAS and PDEFAT calls right after the PINCL call.

PDEFF calls should be next. The PDEFF calls define the independent and dependent variable by user name. The final argument in the PDEFF call, is not in quotes. It does not have to be the same as the first argument name, as depicted here. However this simplifies programming. If a different final argument is used, its name must match the name of the argument in the DIMENSION and READ statements. For example, if the name FUNCL is used as the last argument instead of CL, then the DIMENSION statement must be

```
DIMENSION FUNCL(140), CM(140)
```

and the READ command must be

```
READ(1,*) FUNCL(I)
```

The next set of commands defines the functions in terms of their function groups. In this example, CL and CM constitute the first function group. Each function group definition sequence begins with a PINITFG(n) call where n is the function group number. The PGENFn calls associate the user's function name in with the internal FGS 300 names. The FGX(n) and FGF(m) definitions are declared explicitly in the subroutine FGEXEC.

The TYPE commands are not required. They are helpful, however, if large tables of data are used. The TYPE commands inform the user that the subroutine is being executed and the SIMSTAR is not locked up.

Memory Overload

In some cases, the SIMSTAR's active memory is unable to support the loading of the FGS 300 as shown. Using the DIMENSION statements as shown causes the SIMSTAR to store some of the data within its own memory. This is not necessary for program execution because once the data is loaded into the FGS 300, the SIMSTAR does not need to access it en masse again. Furthermore, such overloading can cause memory allocation problems resulting in important memory locations being overwritten. The user can examine values of declared constants following a simulation run to determine if this problem has occurred. The values will be incorrect.

To prevent memory allocation problems, SIMSTAR's extended memory may be used to load function data into the FGS 300. Then the numerical values used in the active SIMSTAR program are not disturbed.

To use extended memory, replace the DIMENSION statments with PARAMETER, EXTENDED BLOCK, INTEGER NGET, and CALL X:GDSPCE statements as shown in Example A.3. To determine a minimum value for the NASK argument, divide

the total number of data points to be loaded by 1048. Usually, only the data points need to be loaded into extended memory, as the number of breakpoints is small. If necessary, the breakpoints may be loaded into extended memory also.

Subroutine FGEXEC. The user must write a subroutine to provide explicit mapping of the SIMSTAR's hybrid program's variable and function names associated with the FGS 300's internally-used names. Example A.4 shows the SUBROUTINE FGEXEC for the sample program.

The structure of FGEXEC is simple. The first line following the opening title line is blank except for a plus sign in the first column. The second line is the same INCLUDE statment used in PREINIT.

The next lines equivalence the FGS 300 internal FGX(n) labels to independent variables used by the SIMSTAR's hybrid program. These labels should match those in the argument lists of the PGENFn calls.

After the FGX(n) statements, each function group is invoked with calls to FGS subroutine PEXECFG(n).

Finally, the dependent variables are equivalenced to FGS 300 function names, FGF(m). Again, these should match the FGF(m) assignments made in PREINIT.

The Main Program. Calls to PREINIT and FGEXEC are made in the SIMSTAR main program. As stated earlier,

PREINIT need not be called explicitly. FGEXEC is called twice. Refer to Example A.5.

FGEXEC is called first in the INITIAL region of the main program. It is called again in the derivative region, this time within a PROCEDURAL region.

As shown in the example, the PROCEDURAL call is opened with an argument list which includes all the dependent functions on the left side of an equals sign. All the independent variables are on the right side of the equality sign. The call to FGEXEC is made, and the PROCEDURAL block is closed.

While the SIMSTAR main program is running, FGEXEC automatically accesses the FGS 300. Current independent variables are sent to the FGS 300, and updated function values returned in a manner of microseconds.

Example A.1: Sample Subroutine to Load Function Generator

```
        SUBROUTINE PREINIT
      INCLUDE '@SYSTEM^(SYSTEM)PFGSIN'
C  VARIABLES AOA, MACH SET UP AS GROUP I INPUT ARGUMENTS
C  VARIABLE CL, CM ARE SET UP AS GROUP I FUNCTIONS
C
C  VARIABLES MII, H1 SET UP AS GROUP II INPUT ARGUMENTS
C  VARIABLES RMDH, RYDR SET UP AS GROUP II FUNCTIONS
C
C  DIMENSION BREAKPOINTS, FUNCTIONS
C
C BP1ARY REPRESENTS AOA BREAKPOINTS
C BP2ARY REPRESENTS MACH BREAKPOINTS
      DIMENSION BP1ARY(20), BP2ARY(7)
      DIMENSION CL(140), CM(140)
C
C BP3ARY REPRESENTS MII BREAKPOINTS
C BP4ARY REPRESENTS H1 BREAKPOINTS
      DIMENSION BP3ARY(19) BP4ARY(4)
      DIMENSION RMDH(76), RYDR(76)
C
C LIST BREAKPOINTS
     DATA BP2ARY/.2,.6,.8,.9,1.2,1.6,2.0/
    DATA BP4ARY/0,10,25,50/
C
C DESIGNATE WHERE THE DATA IS
```

```
      OPEN(1,FILE = 'AC.DAT',BLOCKED = .TRUE.)
C
C READ THE DATA INTO THE FGS 300
      DO I = 1, 140
        READ(1,*) CL(I)
     END DO
      DO J = 1, 140
        READ(1,*) CM(J)
      END DO
      TYPE *,'GROUP I DATA READ'
      DO K = 1, 76
        READ(1,*) RMDH(K)
      END DO
      DO L = 1, 76
        READ(1,*) RYDR(L)
      END DO
      TYPE *,'GROUP II DATA READ'
      CLOSE(1)
C
      TYPE *,'ALL DATA DIMENSIONED AND READ'
C ALLOCATE FUNCTION GENERATOR
      CALL PINCL(0,ISTAT)
      IF (ISTAT.NE.0)THEN
       TYPE *,'ALLOCATION OF FGS UNIT 0 DENIED'
       TYPE *,'ISTAT  =  ',ISTAT
      STOP
```

```
      END IF
C
C DEFINE DUMMY ARGUMENTS AND FUNCTIONS
C
      CALL PDEFAS('AOA',-4.0,34.0,20)
      CALL PDEFAT('MACH',BP2ARY,7)
      CALL PDEFAS('MII',0.2,2.0,19)
      CALL PDEFAT('H1',BP3ARY,4)
      TYPE *,'PDEFAS, PDEFAT'
C
C
C DEFINE FUNCTIONS ACCORDING TO SIMSTAR NAMES
C
C
     CALL PDEFF2('CL','AOA','MACH',CL)
      CALL PDEFF2('CM','AOA','MACH',CM)
      TYPE *,'GROUP I PDEFF2'
      CALL PDEFF2('RMDH','MII','H1',RMDH)
      CALL PDEFF2('RYDR','MII','H1',RYDR)
      TYPE *,'GROUP II PDEFF2'
C
C
C DEFINE FUNCTION GROUPS
C
      CALL PINITFG(1)
      CALL PGENF2('CL',FGX(1),FGX(2),FGF(1))
```

```
      CALL PGENF2('CM',FGX(1),FGX(2),FGF(2))
      CALL PTERMFG(1)
C
      TYPE *,'GROUP I PINITFG, PGENF2, PTERMFG'
C
      CALL PINITFG(2)
      CALL PGENF2('RMDH',FGX(3),FGX(4),FGF(3))
      CALL PGENF2('RYDR',FGX(3),FGX(4),FGF(4))
      CALL PTERMFG(2)
C
      TYPE *,'GROUP II PINITFG, PGENF2, PTERMF'
C
      RETURN
      END
```

Example A.2: Data File Format for AC.DAT

CL = f(AOA, MACH)

CM = f(AOA, MACH)

where

AOA= -4 to +34 degrees in 2 degree increments

MACH = 0.2, 0.6, 0.8, 0.9, 1.2, 1.6, 2.0

and

RMDH = f(H1, MII)

RYDR = f(H1, MII)

where

H1 = 0, 10,000, 25,000, 50,000 feet

MACH = 0.2 to 2.0 in 0.1 increments

Data File Structure

CL(MACH = 0.2, AOA = -4)

CL(0.2, -2)

CL(0.2, 0)

...

CL( .2, 34)

CL(0.6, -4)

CL(0.6, -2)

...

CL(0.6, 34)

```
...
CL(2.0, 34)

CM( .2, -4)
...
CM(2.0, 34)

RMDH(H1 = 0, MII = .2)
RMDH(0, 0.3)
RMDH(0, 0.4)
...
RMDH(0,2.0)
RMDH(10000, 0.2)
...
RMDH(10000, 2.0)
...
RMDH(50000,2.0)
```

Example A.3: PREINIT Using SIMSTAR's Extended Memory

```
      SUBROUTINE PREINIT
C  VARIABLES AOA, MACH SET UP AS GROUP I INPUT ARGUMENTS
C  VARIABLE CL, CM ARE SET UP AS GROUP I FUNCTIONS
C
C  VARIABLES MII, H1 SET UP AS GROUP II INPUT ARGUMENTS
C  VARIABLES RMDH, RYDR SET UP AS GROUP II FUNCTIONS
C
C  DIMENSION BREAKPOINTS, FUNCTIONS
C
C BP1ARY REPRESENTS AOA BREAKPOINTS
C BP2ARY REPRESENTS MACH BREAKPOINTS
C BP3ARY REPRESENTS MII BREAKPOINTS
C BP4ARY REPRESENTS H1 BREAKPOINTS
      DIMENSION BP1ARY(20), BP2ARY(7)
      DIMENSION BP3ARY(19) BP4ARY(4)
      PARAMETER(NASK=2)
      EXTENDED BLOCK/G1/CL(140),CM(140),RMDH(76),RYDR(76)
      INTEGER NGET
      CALL X:GDSPCE(NASK,NGET,,)
C
C LIST BREAKPOINTS
     DATA BP2ARY/.2,.6,.8,.9,1.2,1.6,2.0/
    DATA BP4ARY/0,10,25,50/
C
C DESIGNATE WHERE THE DATA IS
```

```
      OPEN(1,FILE = 'AC.DAT',BLOCKED = .TRUE.)
C READ THE DATA INTO THE FGS 300
      DO I = 1, 140
        READ(1,*) CL(I)
      END DO
      DO J = 1, 140
        READ(1,*) CM(J)
      END DO
      TYPE *,'GROUP I DATA READ'
      DO K = 1, 76
        READ(1,*) RMDH(K)
      END DO
      DO L = 1, 76
        READ(1,*) RYDR(L)
      END DO
      TYPE *,'GROUP II DATA READ'
      CLOSE(1)
C
      TYPE *,'ALL DATA DIMENSIONED AND READ'
C ALLOCATE FUNCTION GENERATOR
      CALL PINCL(0,ISTAT)
      IF (ISTAT.NE.0)THEN
       TYPE *,'ALLOCATION OF FGS UNIT 0 DENIED'
       TYPE *,'ISTAT  =  ',ISTAT
      STOP
      END IF
```

```
C DEFINE DUMMY ARGUMENTS AND FUNCTIONS
C
      CALL PDEFAS('AOA',-4.0,34.0,20)
      CALL PDEFAT('MACH',BP2ARY,7)
      CALL PDEFAS('MII',0.2,2.0,19)
      CALL PDEFAT('H1',BP3ARY,4)
      TYPE *,'PDEFAS, PDEFAT'
C
C
C DEFINE FUNCTIONS ACCORDING TO SIMSTAR NAMES
C
C
     CALL PDEFF2('CL','AOA','MACH',CL)
      CALL PDEFF2('CM','AOA','MACH',CM)
      TYPE *,'GROUP I PDEFF2'
      CALL PDEFF2('RMDH','MII','H1',RMDH)
      CALL PDEFF2('RYDR','MII','H1',RYDR)
      TYPE *,'GROUP II PDEFF2'
C
C DEFINE FUNCTION GROUPS
C
      CALL PINITFG(1)
      CALL PGENF2('CL',FGX(1),FGX(2),FGF(1))
      CALL PGENF2('CM',FGX(1),FGX(2),FGF(2))
      CALL PTERMFG(1)
C
```

```
      TYPE *,'GROUP I PINITFG, PGENF2, PTERMFG'
      CALL PINITFG(2)
      CALL PGENF2('RMDH',FGX(3),FGX(4),FGF(3))
      CALL PGENF2('RYDR',FGX(3),FGX(4),FGF(4))
      CALL PTERMFG(2)
C
      TYPE *,'GROUP II PINITFG, PGENF2, PTERMF'
C
      RETURN
      END
```

## Example A.4: Executable Call to Function Generator

```
      SUBROUTINE FGEXEC
+
       FGX(1) = AOA
       FGX(2) = MACH
       FGX(3) = MII
       FGX(4) = H1
C
       CALL PEXECFG(1)
       CALL PEXECFG(2)
C
       CL = FGF(1)
       CM = FGF(2)
       RMDH = FGF(3)
      RYDR = FGF(4)
C
       RETURN
       END
```

Example A.5: FGS 300 Calls Within the SIMSTAR Main Program

```
*PSP=1,0,ERR=ALL
*TITLE
*USE OF THE FGS 300 FUNCTION GENERATOR
*INPUT
PROGRAM
   INITIAL
      CALL FGEXEC
   END $'OF INITIAL'
   DYNAMIC
       DERIVATIVE
            PROCEDURAL(CL,CM,RMDH,RYDR=AOA,MACH,MII,H1)
               CALL FGEXEC
            END $'OF PROCEDURAL'
         '@PARALLEL'
         '@ENDPARALLEL'
      END $ 'OF DERIVATIVE'
   END $ 'OF DYNAMIC'
 TERMINAL
 END $ 'OF TERMINAL'
END $ 'OF PROGRAM'
*TRANSLATE
*OUTPUT
*END
```

# Appendix B: SIMTACS-RT User's Guide

## Overview

SIMTACS-RT was designed to be an easy-to-use tool to test a control law. This appendix will lead the user through the steps necessary to test a controller, and familiarize him with the SIMTACS-RT software structure.

## The Software

There are main programs ready to run on SIMTACS-RT; FEP, MANLO, and TABLE. FEP is designed to use idealized inputs to test a control law, MANLO is designed for man-in-the-loop testing of a longitudinal control law, and TABLE is a short program to extract function values from the aerodynamic data tables.

While the code is not included here due to bulk, it is available in the SIMSTAR laboratory, and should be referenced for this discussion. This document is intended to assist in the use of the code as it exists. To make major modifications requires understanding the D-TRAN and P-TRAN programming languages. Details of the overall program structure is also described in the EAI programming manuals(18).

While detailed knowledge of SIMSTAR's operating procedure and programming languages are not a pre-requisite to using SIMTACS-RT, a familiarity with them is

assumed. The user should understand which parts of the program control the digital and analog processors, and how to identify code used to pass variable values between them.

FEP

The first program, FEP, is simple to use. First load the FEP, ensuring three files, FEP, C.FEP, and P1.FEP are in SIMSTAR's working directory. Ensure also that the file AC1.DAT is in the same directory for function generator access.

Type

FEP

then SIMTACS-RT will load the function generator and the executable SIMSTAR program. Loading the function generator takes approximately two minutes, and SIMSTAR's preparation of the analog processor takes approximately two minutes. During the function generator loading, messages will appear periodically providing information about the data loading status. If no message appears within one minute after typing the above command, check the SIMSTAR and FGS 300 to ensure they are powered and operating correctly. If a message appears,

"REQUIRED ASSET NOT AVAILABLE"

then one of the files listed above is not in SIMSTAR's working directory.

When loading is complete, the

>

prompt will appear, and may be accompanied by

*** MACRO IN OVERLOAD ***

messages. These messages are not a problem if they occur at load time.

Type

>S CMD=10

to activate the command file. At this point it is helpful to refer to a program listing. Lines 15 through 40 of the program source code, S1.FEP, list parameters you will need to set, along with comments as to their function. The command file, C.FEP, short, special-use single-letter or single-word commands to expedite simulation runs.

To use FEP, you will need to set the following parameters.

>S AOA0=1.9 (initial angle of attack value in degrees)

>S PITCH0=1.9 (initial pitch value in degrees)

>S DHTR=-2.8 (initial elevator deflection in degrees)

>S DTZ = 1, MTZ=1, LTZ=1

LTZ, MTZ, DTZ, ITZ, YTZ, NTZ are all switches which turn certain aerodynamic equations on and off. Setting these values to 1 activates the aerodynamic equations. Setting them to 0 deactivates the equations.

>S DFON=1 (This activates the digital flight control law, but does not necessarily give the DFCS command of the aircraft.)

>S ANSW=1 (This gives control of the aircraft to the analog flight control routine. Set ANSW = 0 if the digital controller is to be used.)

>S DGSW=1 (This gives control of the aircraft to the DFCS. Set DGSW=0 if the analog controller is to be used.)

>S AZOSW=1 (to enable the simulated controllers to use transverse acceleration feedback.)

>S NCYCLE=1 (This will run the DFCS at the same computational cycle rate as the rest of the digital processor.)

Set NCYCLE to 3 to run the DFCS at one-third the digital processor rate, etc.

>S DHZ=1 (This activates the horizontal stabilizer.)

>S RUNTIM=15 (where RUNTIM is the length of time in seconds you wish to run the simulation.)

For trim, use the the following settings.

>S ETIN=-3,ET=-3 (when the digital control law is activated.)

>S ETIN=1.05,ET=1.05 (when the analog control law is activated.)

These are the minimum settings necessary to start the simulation of a stable aircraft. Most of these settings are available through command files. Simply typing in the following abbreviated commands will provide the same set-up procedures just listed.

>DHN

will activate the elevator control. This replaces the setting for DHZ.

>MK

will set initial values for pitch, aoa, and elevator deflection. This replaces set commands for PITCH0, AOA0, DHTR.

>AA

will activate equations calculating coefficients of drag, lift, longitudinal moment as required for longitudinal simulation. AA replaces the set commands for LTZ, MTZ, DTZ.

>DFN

will activate the DFCS. This replaces the set command for DFON.

>PREP

will load initial aerodynamic coefficient values, and reset the analog processor to initial conditions.

>T10

sets a runtime of ten seconds.

>T30

sets a runtime of 30 seconds. These last two commands replace the set RUNTIM command.

Other required parameters are preset within SIMTACS-RT code. Examination of the S1.FEP AND C.FEP code will enable the user to exploit additional flexibility by changing variables from the keyboard.

The strip chart recorder sits adjacent to the SIMSTAR terminals and provides a means of monitoring data. From left to right, it records analog elevator deflection (degrees), input force (pounds), angle of attack (radians), pitch (radians), pitch rate (radians per second), the derivative of pitch rate (radians per second per second), G-loading (feet per second per second), and altitude (feet). Precise values of analog variables at the end of a run can be viewed by the command

```
>D variable name
```

or

```
>D/ALL.
```

To run the simulation, once the above variables are set,

type

```
>START
```

Turn on the strip chart recorder if analog data is required.

To log digital data, use the command

```
>PREPAR variable1,variable2
```

This data may be examined using the command

```
>PRINT.
```

If the prepared command file C.FEP is used, simply enter

```
>P
```

to see the data. There is also software available on the

Z-248 in the SIMSTAR lab if a hardcopy or plots of the digital data is required.

Using Ideal Inputs FEP is set up to simulate responses to ideal inputs. To use these inputs, set the following parameters.

>S TDLY1 = 2 (pulse input start time in seconds)

>S TDLY2 = 5 (pulse input end time in seconds)

>S FEGA = 3 (pulse magnitude in pounds)

>S FEGA1= 3 (control stick gain)

Notice both FEGA and FEGA1 determine pulse magnitude. This is because they represent cascaded gains at various points within the flight control system. The designer should scale these to meet his needs, and to correspond to realistic values of stick force.

The code for these ideal inputs resides in the parallel region of S1.FEP immediately below the flight control law. (The flight control law is placed in an asterisk "box" to make it easier to find.) Examination shows that code is also provided to test harmonic inputs for frequency response testing. This code is "commented out" (rendered inactive) to simplify the program and conserve MCB's. To use the harmonic inputs, one must edit S1.FEP, remove the apostrophes around those lines of code, and "comment out" the three lines of code above by placing apostrophes around them. Then the source file, S1.FEP, is recompiled. The newly-generated files, P1.FEP and FEP at

runtime are used at runtime. Input will be controlled by setting input magnitude with the parameter FEGB and frequency with the parameter FREQ.

MANLO

Manlo is almost identical to FEP except that MANLO allows use of the force stick for pilot inputs, and the oscilloscope for video output. The only difference at runtime is that FEGA is not used to control an ideal input magnitude. FEGA1 is still the input gain to the controller. There are commands included in the parallel code to provide a pitch command bar during runtime. TDLY1 and TDLY2 are the variables which control the beginning and end of the pitch-up command, with PMAG controlling the magnitude in degrees.

The force stick and oscilloscope connections are made to SIMSTAR's analog processor. Statements in the TRANSLATE/OUTPUT region of the code lists the connection terminals.

Modifying the Longitudinal Controller

SIMTACS-RT's program code is written modularly to allow an engineer to use his own control law. An analog control law and a digital control law are already in place, and it is recommended that the engineer first familiarize himself with SIMTACS-RT using these controllers. Frequent reference to the simplified schematics (Figure 3.6 and Figure 3.7) is useful.

It is likely that adequate design simulation can be performed by modifying the control laws in place. The advantage to using SIMTACS-RT's in-place controllers is that all code required to interface the SIMSTAR's analog and digital processors, as well as the FGS 300 and DFCS subroutines is ready to use. It is highly recommended that the SIMTACS-RT's code structure be preserved as much as possible for this reason. As the user becomes more familiar with the details of SIMSTAR programming, he can attempt more extensive program modifications.

If modifications are made special attention should be paid to the following "tricky" problems.

1. If additional variables are passed to or from the DFCS, there are three argument lists to update. First, the argument list in the call to SUBROUTINE DFCS and the subroutine argument list itself must match. The argument list for the PROCEDURAL BLOCK code surrounding the DFCS call must be updated as well. In this argument list, variables being fed to the flight control subroutine must follow the equality symbol, and variables returning from the DFCS must precede the equality symbol. In addition, ensure the variables are added to the REAL declarations in the DFCS subroutine code and in the INITIAL region of the SIMSTAR main program.

2. If additional variables are passed from the analog processor to the digital processor, update the PREP1

subroutine, and the BLOCKDATA PREPBD subroutine. If additional variables are passed from the digital processor to the analog processor, update SUBROUTINE POST1, SUBROUTINE BLOCKDATA POSTBD, and SUBROUTINE PREPDCA.

3. Scaling is required for any variables passed to or from the analog processor, whether they come from external inputs, the digital processor, or are sent to external outputs. My experience indicates the MINVAL and MAXVAL statements work better than SCALE statements. Use SCALE statements only to scale integrators.

4. The FGS 300 has capacity for more data. Refer to APPENDIX A for programming instructions. Remember to update the PROCEDURAL argument list surrounding the FGEXEC call.

TABLE

TABLE is a program used to load and access the FGS 300. The code is simple. Inital values for the input variables are set for each run, as well as incremental changes. The resulting data can be logged and compared to data tables to ensure the data has been properly loaded into the FGS 300. TABLE is also useful to examine regions of a data table for linearity and continuity.

More Help

Within the lab there is additional documentation on the various aspects of SIMSTAR programming and runtime operation. One document "Things EAI Never Told You"

contains hints about peculiarities that have not been documented or are not widely known. If stuck, consult this file.

# Bibliography


1. Arpasi, Dale J. and Richard A. Blech. Hardware for a Real-time Multiprocessor Simulator," Distributed Simulation 1985, 43-49. La Jolla, CA: Simulation Councils, Inc., 1985.

2. Barfield, Finley, Expert engineer on the F-16 and AFTI/F-16. Lecture given to EENG 641 class. Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 2 May 1988.

3. Blakelock, John H. Automatic Control of Aircraft and Missiles. New York: John Wiley and Sons, Inc., April 1965.

4. Clapp, Robert E. "Cue Coordination in Training Simulators," Simulators IV--Proceedings of the SCS Simulators Conference, 1987. 217-222. San Diego: Simulation Councils Inc., 1987.

5. D'Azzo, John J. and Constantine H. Houpis. Linear Control Systems, Analysis and Design. New York: McGraw-Hill Book Company, 1987.

6. Etkin, Bernard. Dynamics of Atmospheric Flight. New York: John Wiley & Sons, Inc., 1972.

7. Etkin, Bernard. Dynamics of Flight, Stability and Control. John Wiley & Sons, Inc., 1982.

8. Gunston, Bill. Grumman X-29. Vista, CA: Aeolus Publishing, Limited, 1985.

9. Harrison, Greg, Expert engineer on simulation graphics at the Air Force Flight Dynamics Laboratory. Discussions March 1988 through May 1988.

10. Houpis, Constantine, H. Quantitative Feedback Theory (QFT) Technique for Designing Multivariable Control Systems. AFWAL-TR-86-3107. Flight Dynamics Laboratory, Wright-Patterson AFB, OH. January 1987

11. Houpis, C. H. and Gary B. Lamont. Digital Control Systems Theory, Hardware, Software. New York: McGraw-Hill Book Company, 1985.

12. Kassan, Mark W. 2Lt F-16 Simulator for Man-In-The-Loop Testing of Aircraft Control Systems (SIMTACS), MS Thesis AFIT/GE/ENG/87D-30. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1987 (AD-A189675).

13. Larimer, S. J. An Interactive Computer-Aided Design Program for Digital and Continuous System Analysis and Synthesis (TOTAL), MS thesis, AFIT/GE/GGC/EE/78-2. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 1978; available from Defense Documentation Center (DDC), Cameron Station, Alexandria, VA 22314.

14. Maybeck, Peter S. Stochastic Models, Estimation, and Control, Volume 1. New York: Academic Press, 1982.

15. Milam, Lt Col. Vice Chairman of National Aerospace Plane project. Lecture delivered to EENG 641 class. Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 9 May 1988.

16. Roskam, Jan. Flight Dynamics of Rigid and Elastic Airplanes Lawrence, KA: Roskam Aviation and Engineering Corporation, 1972.

17. Shinn, Capt Thomas B. and Frederick E. Unfried. "Evaluation of the AFTI/F-16 on the LAMARS (Large Amplitude Multi-Mode Aerospace Research Simulator)", Aerospace Simulation, 13: 60-66 (February 1984).

18. STARTRAN Scientific Analysis Environment--SIMSTAR Programming Manual, Electronics Associates Inc., 1988.

19. Walker, Robert, Charles Gregory, Jr., Sunil Shah. "MATRIX-X: A Data Analysis, System Identification, Control Design and Simulation Package," Control Systems Magazine, 2: 30-36 (December 1982).

20. Wierzbanowski, Lt Col Ted, Flight test pilot on the X-29. Briefing to EENG 641 Class. Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 27 April 1988.

21. F-16A OHT 'CLEAN' Aerodynamic Data. Obtained from Jim Ruley on magnetic tape, ASD/ENFTC, Wright-Patterson AFB OH.

22. Advanced Continuous Simulation Language (ASCL) Reference Manual, Mitchell and Gauthier Associates, 1986.

23. Block 25 F-16 Digital Flight Control System Functional Block Diagram. Data Obtained from Lt Bruce Peet, F-16 System Program Office, Wright-Patterson AFB OH.

24. Ruley, J.M. F-16 Simulation User's Guide Version 1; 12 May 1987, ASD/ENFTC, Wright-Patterson AFB, OH.

25. Function Generation System (FGS 300) User's Guide, Electronics Associates Inc., 1987.

# Vita

Capt Gary G Dameron [redacted] In August 1979 he received a Bachelor of Science Degree in Electrical Engineering from the University of Missouri-Columbia, where he was also commissioned into the United States Air Force through the ROTC program

He worked for a year as a nuclear construction engineer at Union Electric Company's Callaway Nuclear Power Plant in central Missouri. In 1980 he attended Air Force Undergraduate Pilot Training at Vance AFB, Oklahoma, graduating in June, 1981. After attending Combat Crew Training School at Castle AFB, California, he served as a KC-135 copilot and aircraft commander at Fairchild AFB, Washington and at Grand Forks ABF, North Dakota. He entered the School of Engineering, Air Force Institute of Technology in June 1987.

[redacted]

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

# REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

| | |
|---|---|
| 1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED | 1b. RESTRICTIVE MARKINGS |
| 2a. SECURITY CLASSIFICATION AUTHORITY | 3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited. |
| 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE | |
| 4. PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/GE/ENG/88D-8 | 5. MONITORING ORGANIZATION REPORT NUMBER(S) |
| 6a. NAME OF PERFORMING ORGANIZATION: School of Engineering; 6b. OFFICE SYMBOL (If applicable): AFIT/ENG | 7a. NAME OF MONITORING ORGANIZATION |
| 6c. ADDRESS (City, State, and ZIP Code): Air Force Institute of Technology, Wright-Patterson AFB, Ohio 45433 | 7b. ADDRESS (City, State, and ZIP Code) |
| 8a. NAME OF FUNDING/SPONSORING ORGANIZATION: Mark Means, Control Synthesis; 8b. OFFICE SYMBOL (If applicable) | 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER |
| 8c. ADDRESS (City, State, and ZIP Code): FLIGHT DYNAMICS LAB, Wright-Patterson AFB OH 45433 | 10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO. |

11. TITLE (Include Security Classification): See Box 19

12. PERSONAL AUTHOR(S): Gary G. Dameron, BS, Capt, USAF

| 13a. TYPE OF REPORT | 13b. TIME COVERED | 14. DATE OF REPORT (Year, Month, Day) | 15. PAGE COUNT |
|---|---|---|---|
| MS Thesis | FROM _____ TO _____ | 1988 December | 123 |

16. SUPPLEMENTARY NOTATION: Fr. Brock

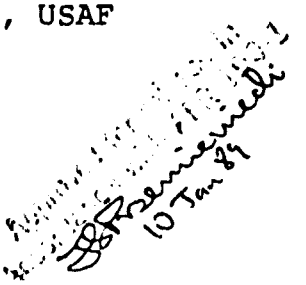
| 17. COSATI CODES | | |
|---|---|---|
| FIELD | GROUP | SUB-GROUP |
| 01 | 04 | |
| | | |

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Flight Control Systems, Flight Simulation, Hybrid Computers, Man-in-the-Loop, Human Factors Engineering, Fighter Aircraft, Theses. (SDC)/JFS).

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: A REAL-TIME SIMULATOR FOR MAN-IN-THE-LOOP TESTING OF AIRCRAFT CONTROL SYSTEMS (SIMTACS-RT)

Thesis Chairman: Zdzislaw H. Lewantowicz, Lt Colonel, USAF



| | |
|---|---|
| 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS | 21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED |
| 22a. NAME OF RESPONSIBLE INDIVIDUAL: Zdzislaw H. Lewantowicz, Lt Col, USAF | 22b. TELEPHONE (Include Area Code): (513) 255-6913 / 22c. OFFICE SYMBOL: AFIT/ENG |

DD Form 1473, JUN 86 — Previous editions are obsolete. — SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

A real-time, high-fidelity simulator is constructed to model F-16 dynamics and control laws. Built around an Electronics Associates Incorporated (EAI) SIMSTAR hybrid computer, the simulator (SIMTACS-RT) uses non-linear, coupled differential equations for its dynamic model. An EAI FGS 300 function generator is used to access an aerodynamic data base of 25000 values in real time. Man-in-the-loop simulation is supported with a force stick for pilot inputs and an oscilloscope display for pitch and roll information (the lateral program is still in development). Four hybrid computer programs are presented as user-ready simulation/analysis tools, supporting both multi-rate digital and analog control laws. Recommendations for further improvement in simulator realism are presented.